LINEAR PREDICTION ANALYSIS AND SYNTHESIS OF SPEECH SIGNALS

A project work submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
ELECTRONICS AND COMMUNICATION ENGINEERING

by

D. SNEHITH KUMAR **G.G.N.V. KISHORE**

N.PRANATHI **KUMAR ANAND**

Guide

A S N Murthy M.E
Sr. Assistant Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

GITAM INSTITUTE OF TECHNOLOGY

GITAM UNIVERSITY
(Estd. U/S 3 of UGC Act, 1956 & Accredited by NAAC with A Grade)

VISAKHAPATNAM – 530045

April 2013

LINEAR PREDICTION ANALYSIS AND SYNTHESIS OF SPEECH SIGNALS

A project work submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY
IN

ELECTRONICS AND COMMUNICATION ENGINEERING

by

D.SNEHITH KUMAR (1210409113) **G.G.N.V. KISHORE (1210409114)**

N.PRANATHI (1210409133) **KUMAR ANAND (1210409127)**

Guide

A S N Murthy M.E
Sr. Assistant Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

GITAM INSTITUTE OF TECHNOLOGY

GITAM UNIVERSITY
(Estd. U/S 3 of UGC Act, 1956 & Accredited by NAAC with A Grade)

VISAKHAPATNAM – 530045

April 2013

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

GITAM INSTITUTE OF TECHNOLOGY

GITAM UNIVERSITY

(Estd. U/S 3 of UGC Act, 1956 & Accredited by NAAC with 'A' Grade)

VISAKHAPATNAM – 530045

CERTIFICATE

This is to certify that the project work entitled **"LINEAR PREDICTION ANALYSIS AND SYNTHESIS OF SPEECH SIGNALS"** is a bonafide work done by **D.Snehith Kumar** (1210409113), **G.G.N.V.KISHORE** (1210409114), **N.PRANATHI** (1210409133) and **KUMAR ANAND** (1210409127) submitted in partial fulfilment of the requirement for the award of the degree of **Bachelor of Technology** in **Electronics and Communication Engineering** during the Academic year 2012-13.

A.S.N. Murthy

Sr. Assistant Professor
Department of ECE
GITAM University

Dr. V. Malleswara Rao

Professor & HOD
Department of ECE
GITAM University

CONTENTS

Description **Page no.**

ACKNOWLEDGEMENT

I take this opportunity to express my deep sense of gratitude to **A.Satyanarayana Murthy,** Sr. Assistant Professor from the Department of Electronics and Communication Engineering, GITAM University for his valuable guidance in sharing his knowledge and investing time for me throughout this project. I will always remain thankful for his insightful advice and unconditional support.

We are very much pleased to thank our Course Coordinator **Mrs. S.Neeraja**, Assistant professor for her guidance and valuable suggestions while making the documentation and seminar presentations. We convey our sincere thanks to all teaching and non-teaching staff of GIT, for their support and encouragement.

We would also like to convey our sincere thanks to **Dr.V.Malleswara Rao**, Professor & Head of the Department, Electronics and Communication Engineering for his support during this assignment.

Last but not least we would like to acknowledge our indebtedness to all those who devoted themselves directly or indirectly to make this project work a total success.

ABSTRACT

A new approach to speech analysis and synthesis in which we represent the speech waveform directly in terms of time-varying parameters related to the transfer function of the vocal tract and the characteristics of the source functions. By modelling the speech wave itself, rather than its spectrum, we avoid the problems inherent in frequency-domain methods. Spectral analysis is a well-known technique for studying signals; its application to speech signals suffers from a number of serious limitations arising from the non-stationary as well as the quasi periodic properties of the speech wave. In this we describe a procedure for efficient encoding of the speech wave by representing it in terms of time-varying parameters related to the transfer function of the vocal tract and the characteristics of the excitation. The speech wave, sampled at particular frequency, is analyzed by predicting the present speech sample as a linear combination of the previous samples. The predictor coefficients are determined by minimizing the mean-squared error between the actual and the predicted values of the speech samples. Parameters namely, the predictor coefficients, the pitch period, a binary parameter indicating whether the speech is voiced or unvoiced, and the rms value of the speech samples—are derived by analysis of the speech wave, encoded and transmitted to the synthesizer. The speech wave is synthesized as the output of a linear recursive filter excited by either a sequence of quasi periodic pulses or a white-noise source.

Application of this method is efficient transmission and storage of speech signals as well as procedures for determining other speech characteristics such as formant frequencies, bandwidths, the spectral envelope, and the autocorrelation function.

LIST OF FIGURES

List of Tables

CHAPTER 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1 Speech

Speech is the vocalized form of human communication. Speech is simply like any other sound – it's only when we hear it that our brains begin to interpret a particular signal as being speech. There is a famous experiment which demonstrates a sentence of sine wave speech. This presents a particular sound recording made from sine waves. Initially, the brain of a listener does not consider this to be speech, and so the signal is unintelligible. However after the corresponding sentence is heard spoken aloud in a normal way, the listener's brain suddenly 'realises' that the signal is in fact speech, and from then on it becomes intelligible. After that the listener cannot 'unlearn' this fact: similar sentences which are generally completely unintelligible to others will be perfectly intelligible to this listener. Apart from this interpretative behavior of the human brain, there are audio characteristics within music and other sounds that are inherently speech-like in their spectral and temporal characteristics. However speech itself is a structured set of continuous sounds, by virtue of its production mechanism. Its characteristics are very well researched, and many specialised analysis, handling and processing methods have been developed over the years especially for this narrow class of audio signals.

1.2 Structure of speech

A **phoneme** is the smallest structural unit of speech: there may be several of these comprising a single word. Usually we write phonemes between slashes to distinguish them, thus /t/ is the phoneme that ends the word 'cat'. Phonemes often comprise distinctly recognizable **phones** which may vary widely to account for different spoken pronunciations. Two alternative pronunciations of a phoneme are usually the result of a choice between two phones that could be used within that phoneme. In such cases, the alternative phone pair is termed **allophones**. Interestingly, phones which are identical except in their spoken tone can be called **allotones**, something which is very common in Mandarin Chinese, where many phonemes can be spoken with a choice of tone to totally change the meaning of a word. Single or clustered phonemes form units of sound organisation called **syllables** which generally allow a natural rhythm in speaking. Syllables usually contain some form of **initial** sound, followed by a

nucleus and then a **final**. Both the initial and the final are optional, and if present are typically consonants, while the syllable nucleus is usually a vowel. Technically a **vowel** is a sound spoken with an open vocal tract, while a **consonant** is one spoken with a constricted, or partially constricted vocal tract, but as with many research areas, these definitions which are so clear and unambiguous on paper are blurred substantially in practice.

The merging of phonemes and words together is one major difficulty in speech processing– especially in the field of continuous speech recognition. For simple, single syllable words, the obvious gaps in a waveform plot will correspond to demarcation points, but as the complexity of an utterance increase, these demarcations become less and less obvious, and often the noticeable gaps are mid-word rather than between words. These difficulties have led speech segmentation to being a flourishing research area

Finally, spoken enunciation is context sensitive. When background noise is present we shout, during extreme quiet we whisper. This does not always hold true for communications channels: imagine a man in a quiet office telephoning his wife in a noisy shopping mall. The husband will naturally talk fairly quietly in order not to disturb his colleagues, but the wife will have to shout to be heard. Possibly the wife will ask the office-bound husband to speak up a little and the husband will then ask the wife to stop shouting.

1.3Mechanism of Human Speech Production

In order to better understand the different types of sounds produced by humans, we need to first gain a rudimentary understanding of the mechanisms which produce these sounds in the first place, i.e., the human vocal system. Speech is basically generated when sound pressure waves (typically originating from air pushed out by the lungs) are channeled or restricted in various ways by manual control of the anatomical components of the human vocal system. How the sound wave travels through the entire anatomical system until it is radiated from the lips to the outside world controls the type of sound produced (and heard by a listener). The study and classification of these various sounds is known as Phonetics.

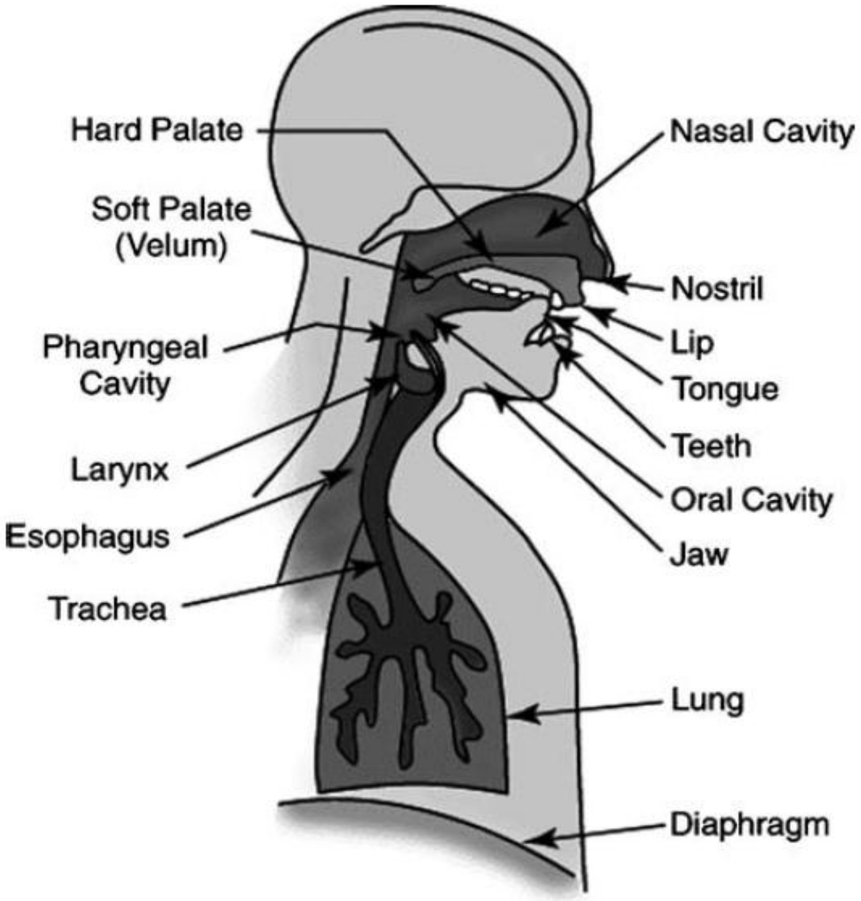


Fig. 1.1 Basic components of the human speech production system

Figure 1.1 illustrates the key anatomical components of the human speech production system. While not delving a detailed biological discussion of this system, let us look at the key portions that affect how speech is produced.

- The sub glottal section of this structure, comprising of the lungs, Bronchi, Trachea, and Esophagus, serves as the original source of the sound energy, i.e., the place where the sound waves emanate.
- The Pharynx, or pharyngeal cavity, is the section from the Esophagus to the mouth.
- The Oral Cavity is the cavity just inside the mouth. Together, the Pharynx and the Oral Cavity (which really forms a composite region where sound pressure waves can travel and be manipulated) are popularly known as the Vocal Tract.
- A critical part of the Larynx is a set of folds of tissue which are collectively known as the Vocal Cords. As we will see shortly, the Vocal Cords play a direct role in controlling the periodicity (pitch) of certain types of speech.

- An alternate path for sound waves to emerge into the ambient air is the cavity leading to the nose, which is known as the Nasal Cavity.
- Apart from the three cavities and vocal cords listed above, other organs such as the lips, teeth, tongue, nostril, and soft palate (Velum) also play a critical role in controlling the specifics of the generated sound.

When air is pushed out of the lungs, the resulting air flow can be subjected to different constrictions and perturbations, all of which together determine what the generated sound would be. From a linguistic perspective, spoken words or syllables are conceptually broken up into sub syllables called Phonemes. For example, the utterance of a single letter might be an example of a Phoneme, though there could certainly be multiple phonemes associated with the same letter when pronounced in different ways and in different languages. The Vocal Tract acts as a filter for the sound waves entering it, effectively creating poles or maxima at a certain frequency (known as Fundamental Frequency) and its harmonics, as shown in above Fig. These pole frequencies are collectively referred to as Formant Frequencies, and the Formant Frequencies uniquely characterize the vocal tract response for a particular voiced sound.

(a) Lung power mostly affects the volume of the sound, but rapid variation often distinguishes a boundary between syllables.

(b) If the glottis is closed temporarily during speech, a glottal stop results such as the /t/ in a Yorkshire-accented reading of ‘I went t’ shops’. A plosive sound like the /d/ in ‘dog’ is a short stop followed by an explosive release.

(c) Vocal chord muscle tension causes the chords to vibrate at different rates, forming the pitch frequencies. Voiceless sounds (e.g. /s/ in ‘six’), where the vocal chords do not vibrate, have little or no pitch structure.

(d) If the air is diverted through the nose by the velum closing, a nasal sound such as /m/ in ‘mad’ results. Different timbre also results from the slightly different path length from lungs to nose compared with lungs to mouth (imagine two different length organ pipes).

(e) If the air travels through the mouth, a humped tongue and opening then closing lower jaw cause a vowel sound (e.g. /a/ in ‘card’), if the lower jaw does not close, a glide (e.g. /w/ in ‘won’) is the result.

(f) Different sounds also result if the air is forced past the sides of a tongue touching the roof of the mouth or the teeth (e.g. /l/ in ‘luck’, and the /th/ sound).

The above actions must be strung together by the speaker in order to construct coherent sentences. In practice, sounds will slur and merge into one another to some extent, such as the latter part of a vowel sound changing depending on the following sound. This can be illustrated by considering how the /o/ sound in ‘or’ and in ‘of’ differ.

1.4Types of Speech Signals

We have already seen the basic structure of the human speech production system. To understand and appreciate the effect of the anatomical structures and movements even further, let us now discuss the main categories of vocal sounds (voiced unvoiced, fricatives, stops, and nasal) and how they are influenced by the interaction of the vocal cords and the vocal tract. As we will see in later chapters, classification of speech sounds is a significant and important problem in speech signal processing algorithms.

1.4.1. Voiced Sounds

Voiced sounds are produced when air flows produced by the lungs are forced through the Glottis on to the vocal cords. The vocal cords consist of folds of tissue stretched across the opening of the Larynx; the tension on these tissues can be adjusted by the person, thereby producing a lateral movement of the vocal cords that cause them to vibrate.

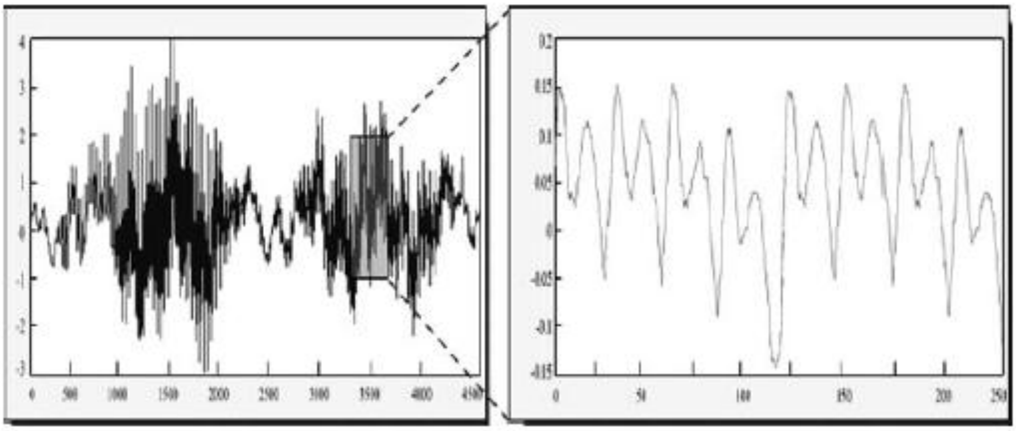


Fig. 1.2 Example of a voiced segment within a speech signal

These vibrations of the vocal cord cause the air flow to exit the Larynx (through a slit between the tissues, known as Glottis) in a roughly periodic or “quasi periodic” pattern. These periodic pulses of air, when not subject to any other major constrictions further down the vocal tract, produce speech at the lips that have strong quasi periodic characteristics. This process of modulation of the pressure wave by vibrating vocal cords is commonly known as Phonation, and the specific characteristics of this modulation are determined not only by the tension applied to the vocal cords but also by the structure and mass of the vocal cords themselves. In summary, voiced sounds are distinguished by the presence of periodicity in the corresponding acoustic waveforms, as shown in the voiced speech segment highlighted in Figure 1.2

The resonant frequency at which the vocal cords vibrate, which is known as the Pitch Frequency, is directly related to the perception of Pitch in voiced speech. This Pitch Frequency is typically in the range of 50–200Hz in the case of male speakers, whereas it can be more than twice as much in the case of female speakers; hence the perception of the average female voice has a “high pitch” relative to the average male voice. In general, voiced speech segments are associated with the vowel sounds of the English language. In some cases, the associated phoneme might consist of not one vowel but two or more vowels occurring successively: these are also known as Diphtongs.

From Fig 1.2 (Or any graphical representations of speech utterances), it should also be apparent that a speech sequence consists of a series of shorter speech segments, and each segment should be analyzed to determine if it contains voiced speech or not. Fortunately, the characteristics of speech signals (i.e., the type of sound) do not change very frequently, and analysis windows in the 20–30ms range are the most common (and in fact standardized in several speech processing industry standards). A Short Time Fourier Transform (or an FFT computation thereof) can be utilized to inspect the changing periodic properties of successive speech segments. As we will see in the context of Voice Coders, successful identification of voiced speech segments is vital to the effectiveness of many algorithms (although this identification might not be made by explicitly computing an FFT: : :but more on that later).

1.4.2. Unvoiced Sounds

Unlike voiced speech, unvoiced sounds do not have any underlying periodicity, and therefore do not have a distinct association with Pitch. In fact, in speech processing algorithms, the unvoiced speech entering the vocal tract is generally modeled as a random white noise source, which turns out to be a fairly good approximation in practical applications. The critical difference between unvoiced and voiced sounds is that in unvoiced sounds there is no significant vibration of the vocal cords (which was the source of the periodicity in voiced speech).

In general, unvoiced sounds are closely related to utterance of consonants in the English language; even intuitively it can be observed that one cannot estimate the pitch of a person's voice by only hearing a consonant-based phoneme being uttered.

1.4.3. Voiced and Unvoiced Fricatives

In a class of phonemes known as Fricatives, the source of the excitation is a partial constriction in the vocal tract, resulting in a localized turbulence and hence noise like properties in the generated speech. Fricatives can be either unvoiced (e.g., /f/ or /sh/) or voiced (e.g., /th/ or /z/). The main difference is that in voiced Fricatives, the noisy characteristics caused by the constriction are also accompanied by vibrations of the vocal cords, thereby imparting some periodicity in the produced sound.

1.4.4. Voiced and Unvoiced Stops

Many unvoiced sounds are produced when there is some kind of constriction in the vocal tract that causes it to be completely closed. For example, when uttering the phoneme /p/ this constriction is at the lips, whereas for /g/ it is at the back of the teeth. In fact, these two examples are part of a category of sounds referred to as Stops. It may be noted that not all Stop sounds are unvoiced; indeed, /b/ and /d/ are examples of voiced Stop sounds. The main difference is that when pressure builds up in the vocal tract due to the constriction, the vocal cords do not vibrate in the case of unvoiced Stops whereas they do vibrate in the case of voiced Stops.

1.4.5. Nasal Sounds

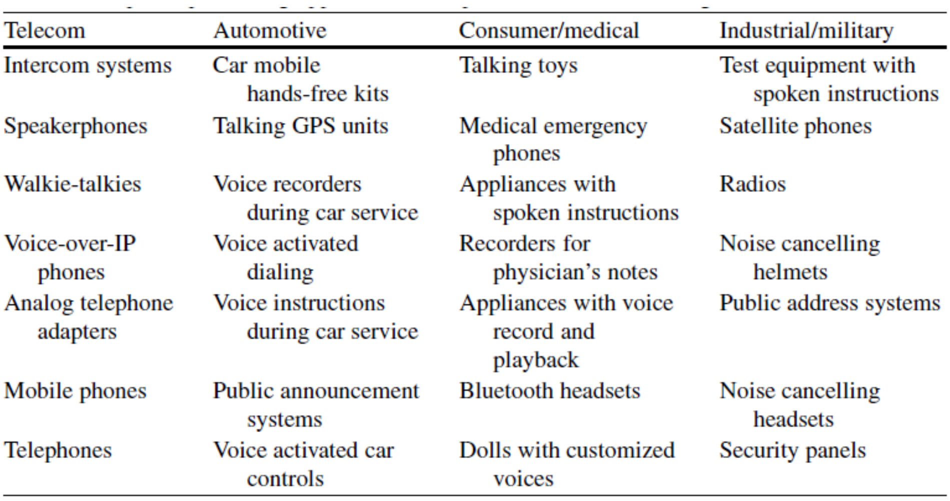
Finally, there are several sounds that are inherently nasal in their characteristics, such as /m/ and /n/. These are generated when the Velum is lowered such that oral cavity is constricted (though still coupled with the Pharynx) and the air pressure flows through the Nasal Tract instead. Here, the mouth acts as a resonant cavity for certain frequencies. The spectral responses for nasal sounds are typically broader (damped) than for voiced sounds due to the coupling of the oral and nasal tracts. Note that there is also greater energy loss due to the intricate structure of the nasal passage.

1.5. Speech Processing in Everyday Life

The proliferation of embedded systems in consumer electronic products, industrial control equipment, automobiles, and telecommunication devices and networks has brought the previously narrow discipline of speech signal processing into everyday life. The availability of low-cost and versatile microprocessor architectures that can be integrated into speech processing systems has made it much easier to incorporate speech-oriented features even in applications not traditionally associated with speech or audio signals.

Perhaps the most conventional application area for speech processing is Telecommunications. Traditional wired telephone units and network equipment are now overwhelmingly digital systems, employing advanced signal processing techniques like speech compression and line echo cancellation. Accessories used with telephones, such as Caller ID systems, answering machines, and headsets are also major users of speech processing algorithms. Speakerphones, intercom systems, and medical emergency notification devices have their own sophisticated speech processing requirements to allow effective and clear two-way communications, and wireless devices like walkie-talkies and amateur radio systems need to address their communication bandwidth and noise issues. Mobile telephony has opened the floodgates to a wide variety of speech processing techniques to allow optimal use of bandwidth and employ value-added features like voice-activated dialing. Mobile hands-free kits are widely used in an automotive environment.

Industrial control and diagnostics is an emerging application segment for speech processing. Devices used to test and log data from industrial machinery, utility meters, network equipment, and building monitoring systems can employ voice prompts and prerecorded audio messages to instruct the users of such tools as well as user-interface enhancements like voice commands. This is especially useful in environments wherein it is difficult to operate conventional user interfaces like keypads and touch screens. Some closely related applications are building security panels, audio explanations for museum exhibits, emergency evacuation alarms, and educational and linguistic tools. Automotive applications like hands-free kits, GPS devices, Bluetooth headsets/helmets, and traffic announcements are also fast emerging as adopters of speech processing. With ever-increasing acceptance of speech signal processing algorithms and inexpensive hardware solutions to accomplish them, speech-based features and interfaces are finding their way into the home. Future consumer appliances will incorporate voice commands, speech recording and playback, and voice-based communication of commands between appliances. Usage instructions could bevocalized through synthesized speech generated from user manuals. Convergence of consumer appliances and voice communication systems will gradually lead to even greater integration of speech processing in devices as diverse as refrigerators and microwave ovens to cable set-top boxes and digital voice recorders.



Telecom	Automotive	Consumer/medical	Industrial/military
Intercom systems	Car mobile hands-free kits	Talking toys	Test equipment with spoken instructions
Speakerphones	Talking GPS units	Medical emergency phones	Satellite phones
Walkie-talkies	Voice recorders during car service	Appliances with spoken instructions	Radios
Voice-over-IP phones	Voice activated dialing	Recorders for physician's notes	Noise cancelling helmets
Analog telephone adapters	Voice instructions during car service	Appliances with voice record and playback	Public address systems
Mobile phones	Public announcement systems	Bluetooth headsets	Noise cancelling headsets
Telephones	Voice activated car controls	Dolls with customized voices	Security panels

Table 1.1 Speech processing application examples in various market segments

Table 1.1 lists some common speech processing applications in some key market segments: Telecommunications, Automotive, Consumer/Medical, and Industrial/ Military. This is by no means an exhaustive list; indeed, we will explore several speech processing applications in the chapters that follow. This list is merely intended to demonstrate the variety of roles speech processing plays in our daily life (either directly or indirectly).

1.6. Common Speech Processing Tasks

Figure depicts some common categories of signal processing tasks that are widely required and utilized in Speech Processing applications, or even general purpose embedded control applications that involve speech signals. Most of these tasks are fairly complex, and are detailed topics by themselves, with a substantial amount of research literature about them. Several embedded systems manufacturers (particularly DSP and DSC vendors) also provide software libraries and/or application notes to enable system hardware/software developers to easily incorporate these algorithms into their end-applications. Hence, it is often not critical for system developers to know the inner workings of these algorithms, and knowledge of the corresponding Application Programming Interface (API) might suffice.

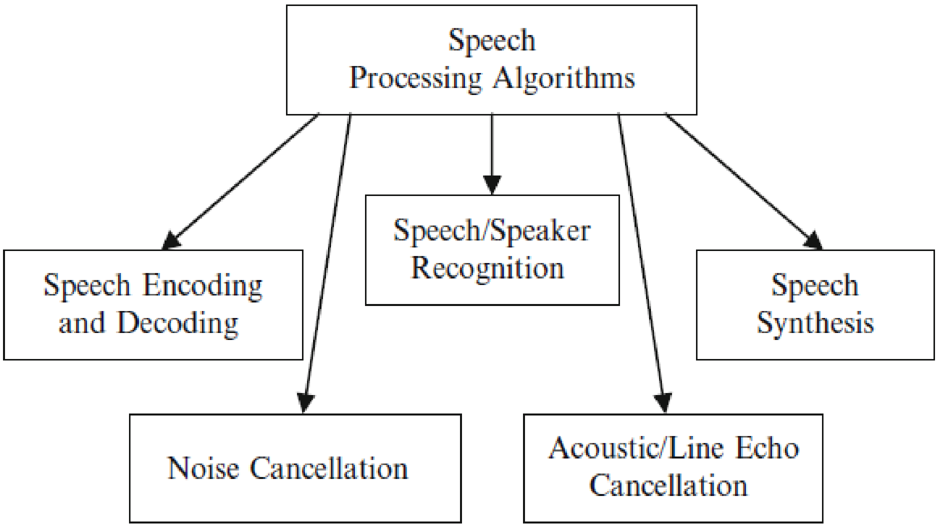

Speech
Processing Algorithms
Speech/Speaker Recognition
Speech Encoding and Decoding
Speech Synthesis
Noise Cancellation
Acoustic/Line Echo Cancellation

Fig. 1.3 Popular signal processing tasks required in speech-based applications

However, in order to make truly informed decisions about which specific speech processing algorithms are suitable for performing a certain task in the application, it is necessary to understand these techniques to some degree. Moreover, each of these speech processing tasks can be addressed by a tremendous variety of different algorithms, each with different sets of capabilities and configurations and providing different levels of speech quality. The system designer would need to understand the differences between the various available algorithms/techniques and select the most effective algorithm based on the application's requirements. Another significant factor that cannot be analyzed without some Speech Processing knowledge is the computational and peripheral requirements of the technique being considered.

CHAPTER 2

LINEAR PREDICTION

CHAPTER 2

LINEAR PREDICTION

2.1. Introduction to LPC

Every cell phone solves 10 linear equations in 10 unknowns every 20 milliseconds Although most people see the cell phone as an extension of conventional wired phone service or POTS (plain old telephone service), the truth is that cell phone technology is extremely complex and a marvel of technology. Very few people realize that these small devices perform hundreds of millions of operations per second to be able to maintain a phone conversation. If we take a closer look at the module that converts the electronic version of the speech signal into a sequence of bits, we see that for every 20 ms of input speech, a set of speech model parameters is computed and transmitted to the receiver. The receiver converts these parameters back into speech.

There exist many different types of speech compression that make use of a variety of different techniques. However, most methods of speech compression exploit the fact that speech production occurs through slow anatomical movements and that the speech produced has a limited frequency range. The frequency of human speech production ranges from around 300 Hz to 3400 Hz. Speech compression is often referred to as speech coding which is defined as a method for reducing the amount of information needed to represent a speech signal. Most forms of speech coding are usually based on a lossy algorithm. Lossy algorithms are considered acceptable when encoding speech because the loss of quality is often undetectable to the human ear.

There are many other characteristics about speech production that can be exploited by speech coding algorithms. One fact that is often used is that period of silence take up greater than 50% of conversations. An easy way to save bandwidth and reduce the amount of information needed to represent the speech signal is to not transmit the silence. Another fact about speech production that can be taken advantage of is that mechanically there is a high correlation between adjacent samples of speech. Most forms of speech compression are achieved by modelling the process of speech production as a linear digital filter. The digital filter and its slow changing parameters are usually encoded to achieve compression from the speech signal.

Linear Predictive Coding (LPC) is one of the methods of compression that models the process of speech production. Specifically, LPC models this process as a linear sum of earlier samples using a digital filter inputting an excitement signal. An alternate explanation is that linear prediction filters attempt to predict future values of the input signal based on past signals. LPC "...models speech as an autoregressive process, and sends the parameters of the process as opposed to sending the speech itself". It was first proposed as a method for encoding human speech by the United States Department of Defence in federal standard 1015, published in 1984. Another name for federal standard 1015 is LPC-10 which is the method of linear predictive coding

Linear predictive coding (LPC) is defined as a digital method for encoding an analog signal in which a particular value is predicted by a linear function of the past values of the signal. It was first proposed as a method for encoding human speech by the United States Department of Defence in federal standard 1015, published in 1984. Human speech is produced in the vocal tract which can be approximated as a variable diameter tube. The linear predictive coding (LPC) model is based on a mathematical approximation of the vocal tract represented by this tube of a varying diameter. At a particular time, t, the speech sample s (t) is represented as a linear sum of the p previous samples. The most important aspect of LPC is the linear predictive filter which allows the value of the next sample to be determined by a linear combination of previous samples. Under normal circumstances, speech is sampled at 8000 samples/second with 8 bits used to represent each sample. This provides a rate of 64000 bits/second. Linear predictive coding reduces this to 2400 bits/second. At this reduced rate the speech has a distinctive synthetic sound and there is a noticeable loss of quality. However, the speech is still audible and it can still be easily understood. Since there is information loss in linear predictive coding, it is a lossy form of compression.

Speech coding or compression is usually conducted with the use of voice coders or vocoders. There are two types of voice coders: waveform-following coders and model-base coders. Waveform following coders will exactly reproduce the original speech signal if no quantization errors occur.

Model-based coders will never exactly reproduce the original speech signal, regardless of the presence of quantization errors, because they use a parametric model of speech production which involves encoding and transmitting the parameters not the signal. LPC vocoders are considered model-based coders which mean that LPC coding is lossy even if no quantization errors occur.

All vocoders, including LPC vocoders, have four main attributes: bit rate, delay, complexity, quality. Any voice coder, regardless of the algorithm it uses, will have to make trade-offs between these different attributes. The first attribute of vocoders, the bit rate, is used to determine the degree of compression that a vocoder achieves. Uncompressed speech is usually transmitted at 64 kb/s using 8 bits/sample and a rate of 8 kHz for sampling. Any bit rate below 64 kb/s is considered compression.

The linear predictive coder transmits speech at a bit rate of 2.4 kb/s, an excellent rate of compression. Delay is another important attribute for vocoders that are involved with the transmission of an encoded speech signal. Vocoders which are involved with the storage of the compressed speech, as opposed to transmission, are not as concern with delay. The general delay standard for transmitted speech conversations is that any delay that is greater than 300 ms is considered unacceptable. The third attribute of voice coders is the complexity of the algorithm used. The complexity affects both the cost and the power of the vocoder. Linear predictive coding because of its high compression rate is very complex and involves executing millions of instructions per second. LPC often requires more than one processor to run in real time. The final attribute of vocoders is quality. Quality is a subjective attribute and it depends on how the speech sounds to a given listener. One of the most common tests for speech quality is the absolute category rating (ACR) test. This test involves subjects being given pairs of sentences and asked to rate them as excellent, good, fair, poor, or bad. Linear predictive coders sacrifice quality in order to achieve a low bit rate and as a result often sound synthetic. An alternate method of speech compression called adaptive differential pulse code modulation (ADPCM) only reduces the bit rate by a factor of 2 to 4, between 16 kb/s and 32kb/s, but has a much higher quality of speech than LPC.

The general algorithm for linear predictive coding involves an analysis or encoding part and a synthesis or decoding part. In the encoding, LPC takes the speech signal in blocks or frames of speech and determines the input signal and the coefficients of the filter that will be capable of reproducing the current block of speech. This information is quantized and transmitted. In the decoding, LPC rebuilds the filter based on the coefficients received. The filter can be thought of as a tube which, when given an input signal, attempts to output speech. Additional information about the original speech signal is used by the decoder to determine the input or excitation signal that is sent

2.2. History of LPC

The history of audio and music compression begin in the 1930s with research into pulse-code modulation (PCM) and PCM coding. Compression of digital audio was started in the 1960s by telephone companies who were concerned with the cost of transmission bandwidth. Linear Predictive Coding's origins begin in the 1970s with the development of the first LPC algorithms. Adaptive Differential Pulse Code Modulation (ADPCM), another method of speech coding, was also first conceived in the 1970s. In 1984, the United States Department of Defence produced federal standard 1015 which outlined the details of LPC. Extensions of LPC such as Code Excited Linear Predictive (CELP) algorithms and Vector Selectable Excited Linear Predictive (VSELP) algorithms were developed in the mid-1980s and used commercially for audio music coding in the later part of that decade. The 1990s have seen improvements in these earlier algorithms and an increase in compression ratios at given audio quality levels.

The history of speech coding makes no mention of LPC until the 1970s. However, the history of speech synthesis shows that the beginnings of Linear Predictive Coding occurred 40 years earlier in the late 1930s. The first vocoder was described by Homer Dudley in 1939 at Bell Laboratories. Dudley developed his vocoder, called the Parallel Band pass Vocoder or channel vocoder, to do speech analysis and resynthesis. LPC is a descendent of this channel vocoder. The analysis/synthesis scheme used by Dudley is the scheme of compression that is used in many types of speech compression such as LPC. The synthesis part of this scheme was first used even earlier than the 1930s by Kempelen Farkas Lovag (1734-1804). He used it to make the first machine that could speak. The machine was constructed using a bellow which forced air through a flexible tube to produce sound.

Homer Dudley and his vocoder

Analysis/Synthesis schemes are based on the development of a parametric model during the analysis of the original signal which is later used for the synthesis of the source output. The transmitter or sender analyses the original signal and acquires parameters for the model which are Path of Human Speech Production sent to the receiver. The receiver then uses the model and the parameters it receives to synthesize an approximation of the original signal. Historically, this method of sending the model parameters to the receiver was the earliest form of lossy speech compression. Other forms of lossy speech compression that involve sending estimates of the original signal weren’t developed until much later.

Because speech signals vary with time, this process is done on short chunks of the speech signal, which are called frames; generally 30 to 50 frames per second give intelligible speech with good compression.

2.3. Overview of LPC

LPC starts with the assumption that a speech signal is produced by a buzzer at the end of a tube (voiced sounds), with occasional added hissing and popping sounds (sibilants and plosive sounds). Although apparently crude, this model is actually a close approximation of the reality of speech production. The glottis (the space between the vocal folds) produces the buzz, which is characterized by its intensity (loudness) and frequency (pitch). The vocal tract (the throat and mouth) forms the tube, which is characterized by its resonances, which give rise to formants, or enhanced frequency bands in the sound produced. Hisses and pops are generated by the action of the tongue, lips and throat during sibilants and plosives.

LPC analyzes the speech signal by estimating the formants, removing their effects from the speech signal, and estimating the intensity and frequency of the remaining buzz. The process of removing the formants is called inverse filtering, and the remaining signal after the subtraction of the filtered modeled signal is called the residue.

The numbers which describe the intensity and frequency of the buzz, the formants, and the residue signal, can be stored or transmitted somewhere else. LPC synthesizes the speech signal by reversing the process: use the buzz parameters and the residue to create a source signal, use the formants to create a filter (which represents the tube), and run the source through the filter, resulting in speech.

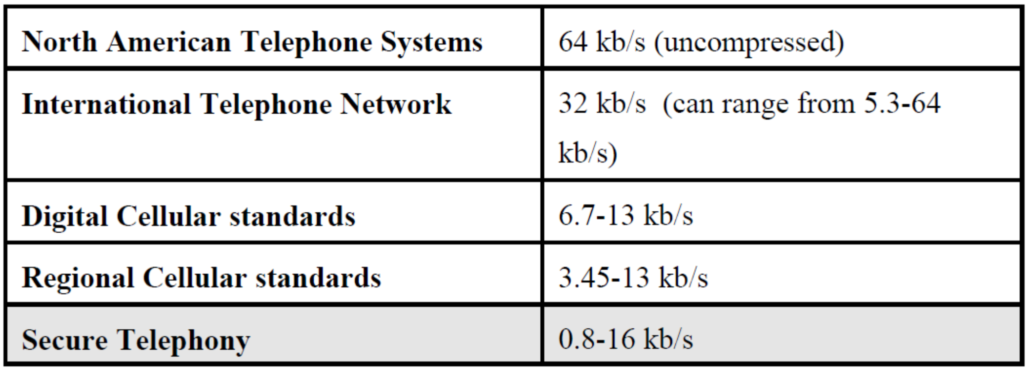
2.4. Applications of LPC

- LPC is generally used for speech analysis and resynthesis. It is used as a form of voice compression by phone companies, for example in the GSM standard. It is also used for secure wireless, where voice must be digitized, encrypted and sent over a narrow voice channel; an early example of this is the US government's Navajo I.
- LPC synthesis can be used to construct vocoders where musical instruments are used as excitation signal to the time-varying filter estimated from a singer's speech. This is somewhat popular in electronic music. Paul Lansky made the well-known computer music piece not just more idle chatter using linear predictive coding. A 10th-order LPC was used in the popular 1980's Speak & Spell educational toy.
- Waveform ROM in some digital sample-based music synthesizers made by Yamaha Corporation may be compressed using the LPC algorithm.
- LPC predictors are used in Shorten, MPEG-4 ALS, FLAC, and other lossless audio codecs.

In general, the most common usage for speech compression is in standard telephone systems. In fact, a lot of the technology used in speech compression was developed by the phone companies. Table 2.1 shows the bit rates used by different phone systems. Linear predictive coding only has 20 applications in the area of secure telephony because of its low bit rate. Secure telephone systems require a low bit rate since speech is first digitalized, then encrypted and transmitted. These systems have a primary goal of decreasing the bit rate as much as possible while maintaining a level of speech quality that is understandable. Other standards such as the digital cellular standard and the international telephone network standard have higher quality standards and therefore require a higher bit rate. In these standards, understanding the speech is not good enough, the listener must also be able to recognize the speech as belonging to the original source.

A second area that linear predictive coding has been used is in Text-to-Speech synthesis. In this type of synthesis the speech has to be generated from text. Since LPC synthesis involves the generation of speech based on a model of the vocal tract, it provides a perfect method for generating speech from text.

Further applications of LPC and other speech compression schemes are voice mail systems, telephone answering machines, and multimedia applications. Most multimedia applications, unlike telephone applications, involve one-way communication and involve storing the data. An example of a multimedia application that would involve speech is an application that allows voice annotations about a text document to be saved with the document. The method of speech compression used in multimedia applications depends on the desired speech quality and the limitations of storage space for the application. Linear Predictive Coding provides a favorable method of speech compression for multimedia applications since it provides the smallest storage space as a result of its low bit rate.



North American Telephone Systems	64 kb/s (uncompressed)
International Telephone Network	32 kb/s (can range from 5.3-64 kb/s)
Digital Cellular standards	6.7-13 kb/s
Regional Cellular standards	3.45-13 kb/s
Secure Telephony	0.8-16 kb/s

Table 2.1 Bit Rates for different telephone standards

2.5. Linear Prediction Properties

Based on the simulations and filter designed in this project we can show some important futures of linear prediction scheme. In the next three sub-sections we discuss these properties.

2.5.1 Decreasing the Dynamic Range

One of the most important futures of linear prediction is decreasing the dynamic range of the residual signal in comparison of the original signal. Fig 2.1 shows the original signal and the residual signal in a single graph. The ability of LP filters in decreasing the dynamic range is clear from this figure.

Another fact about LP filters is that increasing the order of filter decreases the dynamic range of residuals. This fact is shown in Fig 2.2.

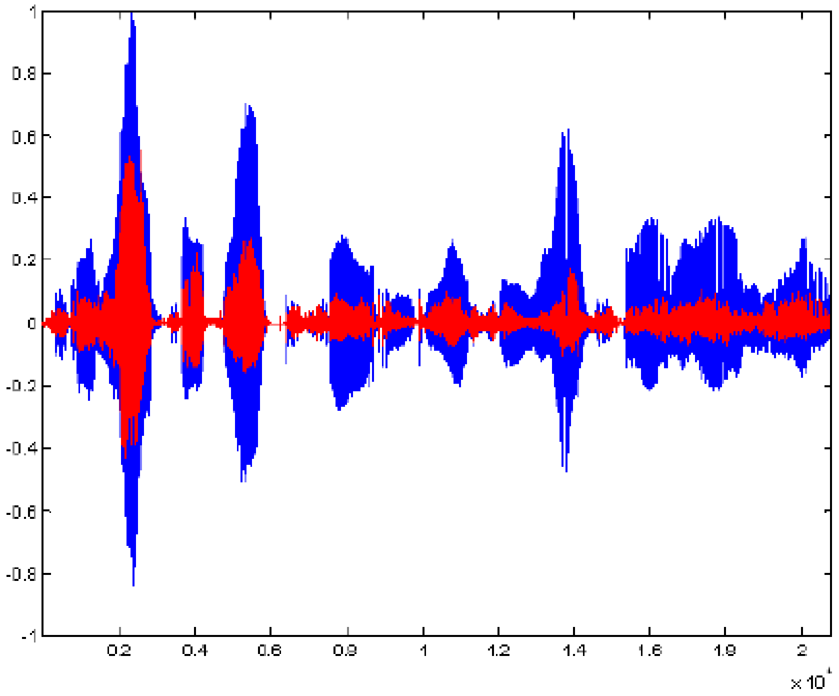


Fig. 2.1 decreasing the dynamic range of residuals.

Blue plot: original signal, Red plot: residuals

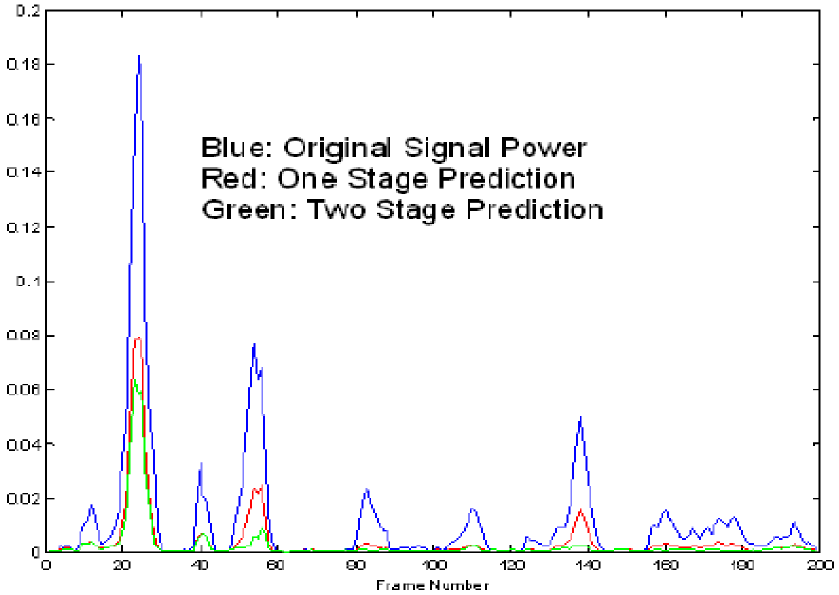


Fig. 2.2 Average power in each frame for original signal and prediction error

When the signal is filtered by a first and second order LP filter

2.5.2 Whitening Property

Another property of linear prediction is that the prediction error is white noise. Fig 2.3 shows the spectrum of original signal and the prediction error. It is clear that although the spectrum of speech signal has most of its energy in high frequencies, the spectrum of residuals is almost flat. This shows that the prediction error is almost white noise.

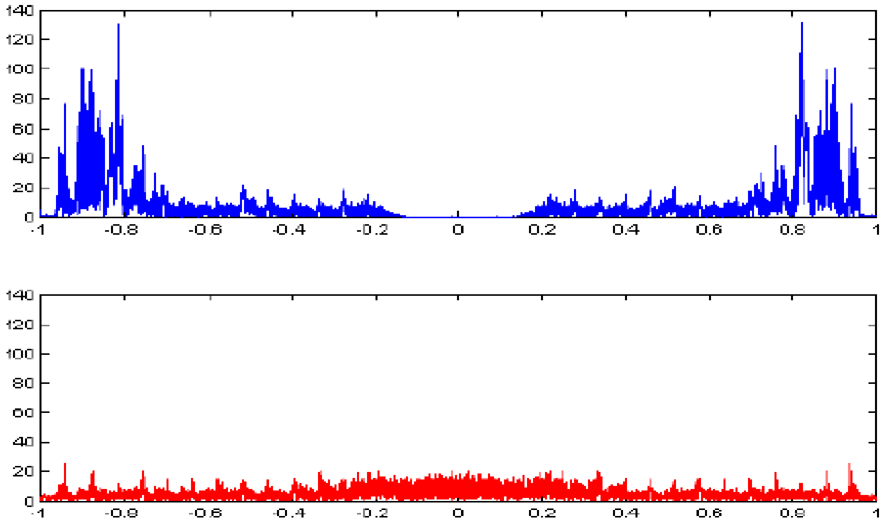


Fig. 2.3 Spectrum of original signal and prediction error.

Top: Original signal, Bottom: prediction error

2.5.3 Stability of Inverse Filter

A FIR linear predictor has its roots inside the unit circle, so its inverse filter is stable. All the poles of LP filters for all frames are given in Fig 2.4. It is clear that all the roots are inside the unit circle.

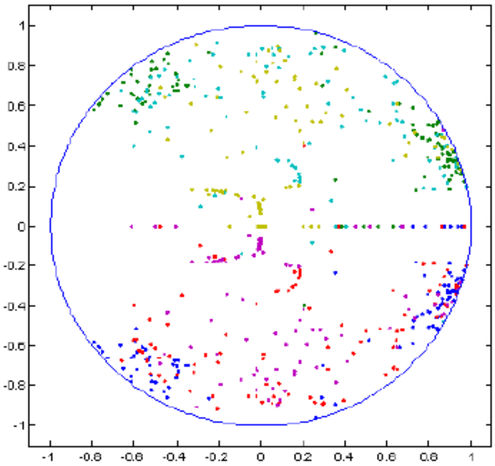


Fig. 2.4 Zeros of LP filters which are the poles of inverse LP filters

CHAPTER 3

LINEAR PREDICTION ANALYSIS

CHAPTER 3

LINEAR PREDICTION ANALYSIS

3.1 LPC Analysis/Encoding

Linear predictive analysis is a simple form of first-order extrapolation: if it has been changing at this rate then it will probably continue to change at approximately the same rate, at least in the short term. This is equivalent to fitting a tangent to the graph and extending the line.

One use of this is in Linear Prediction Analysis which can be used as a method of reducing the amount of data needed to approximately encode a series. Suppose it is desired to store or transmit a series of values representing voice. The value at each sampling point could be transmitted (if 256 values are possible then 8 bits of data for each point are required, if the precision of 65536 levels are desired then 16 bits per sample are required). If it is known that the value rarely changes more than +/- 15 values between successive samples (-15 to +15 is 31 steps, counting the zero) then we could encode the change in 5 bits. As long as the change is less than +/- 15 values in successive steps the value will exactly reproduce the desired sequence. When the rate of change exceeds +/- 15 then the reconstructed values will temporarily differ from the desired value; provided fast changes that exceed the limit are rare it may be acceptable to use the approximation in order to attain the improved coding density.

Speech is produced by an excitation signal generated in our throat, which is modified by resonances produced by different shapes of our vocal, nasal, and pharyngeal tracts. This excitation signal can be the glottal pulses produced by the periodic opening and closing of our vocal folds (which creates voiced speech such as the vowels in “voice”), or just some continuous air flow pushed by our lungs (which creates unvoiced speech such as the last sound in “voice”), or even a combination of both at the same time (such as the first sound in “voice”). The periodic component of the glottal excitation is characterized by its fundamental frequency F_0 (Hz) called pitch1. The resonant frequencies of the vocal, oral, and pharyngeal tracts are called formants. On a spectral plot of a speech frame, pitch appears as narrow peaks for fundamental and harmonics; formants appear as wide peaks of the envelope of the spectrum

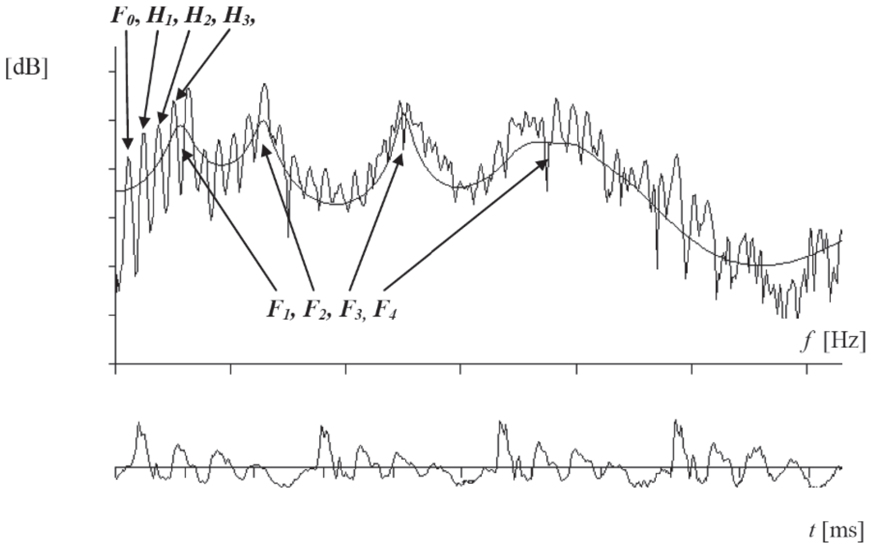


Fig 3.1 denotes a 30- ms frame of voiced speech (bottom) and its spectrum (shown here as the magnitude of its FFT). Harmonics are denoted as H1, H2, H3, etc.; formants are denoted as F1, F2, F3, etc. The spectral envelope is shown here for convenience; it implicitly appears only in the regular FFT

3.2 The LP model of speech

According to government standard 1014, also known as LPC-10, the input signal is sampled at a rate of 8000 samples per second. This input signal is then broken up into segments or blocks which are each analysed and transmitted to the receiver. The 8000 samples in each second of speech signal are broken into 180 sample segments. This means that each segment represents 22.5 milliseconds of the input speech signal.

As early as 1960, Fant proposed a linear model of speech production (Fant 1960), termed as the source-filter model, based on the hypothesis that the glottis and the vocal tract are fully uncoupled. This model led to the well-known autoregressive (AR) or linear predictive (LP) model of speech production (Rabiner and Shafer 1978), which describes speech s (n) as the output $\tilde{s}(n)$ of an all-pole filter 1/A(z) excited by $\tilde{e}(n)$

$$\tilde{S}(z) = \tilde{E}(z)\frac{1}{\sum_{i=0}^{p} a_i z^{-i}} = \tilde{E}(z)\frac{1}{A_p(z)} \qquad (a_0 = 1)$$

Where $\tilde{S}(z)$ and $\tilde{E}(z)$ are the Z transforms of the speech and excitation signals, respectively, and p is the prediction order. The excitation of the LP model (Fig. 3.2) is assumed to be either a sequence of regularly spaced pulses (whose period T0 and amplitude σ can be adjusted) or white Gaussian noise (whose variance σ ² can be adjusted), thereby implicitly defining the so-called voiced/unvoiced (V/UV) decision. The filter $1/A_p(z)$ is termed as the synthesis filter and $A_p(z)$ is called the inverse filter.

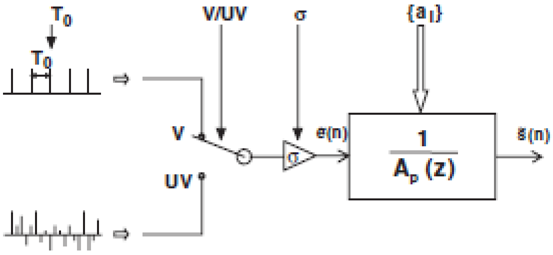


Fig 3.2 The LP model of speech production

Above equation implicitly introduces the concept of linear predictability of speech (hence the name of the model), which states that each speech sample can be expressed as a weighted sum of the p previous samples, plus some excitation contribution:

$$\tilde{s}(n) = \tilde{e}(n) - \sum_{i=1}^{p} a_i \tilde{s}(n-i)$$

3.3 Voice/Unvoiced Determination

According to LPC-10 standards, before a speech segment is determined as being voiced or unvoiced it is first passed through a low-pass filter with a bandwidth of 1 kHz. Determining if a segment is voiced or unvoiced is important because voiced sounds have a different waveform then unvoiced sounds. The differences in the two waveforms create a need for the use of two different input signals for the LPC filter in the synthesis or decoding. One input signal is for voiced sounds and the other is for unvoiced. The LPC encoder notifies the decoder if a signal segment is voiced or unvoiced by sending a single bit. Recall that voiced sounds are usually vowels and can be considered as a pulse that is similar to periodic waveforms. These sounds have high average energy levels which mean that they have very large amplitudes. Voiced sounds also have distinct resonant or formant frequencies. A sample of voiced speech can be seen in Figure 3.3 which shows the waveform for the vowel “e” in the word “test”. Notice that this waveform has the characteristic large amplitude and distinct frequencies of voiced sounds.

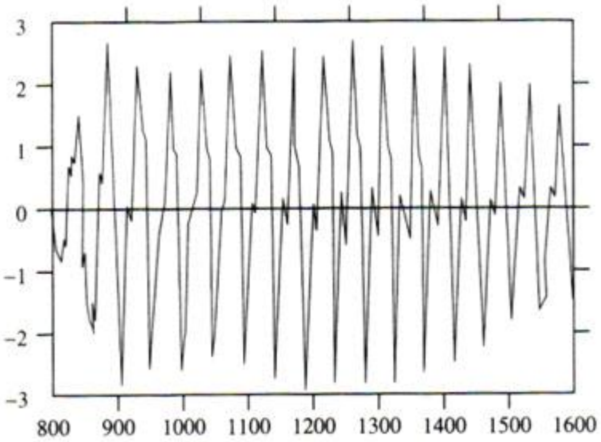


Fig. 3.3 Voiced sound – Letter “e” in the word “test”

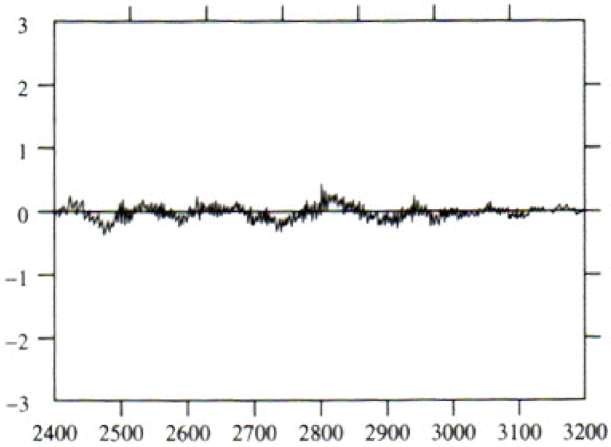


Fig. 3.4 Unvoiced sound – Letter “s” in the word “test”

Unvoiced sounds are usually non-vowel or consonants sounds and often have very chaotic and random waveforms. Figure 3.4 demonstrates that these sounds have less energy and therefore smaller amplitudes then voiced sounds. Figure 3.4 also shows that unvoiced sounds have higher frequencies then voiced sounds.

There are two steps in the process of determining if a speech segment is voiced or unvoiced. The first step is to look at the amplitude of the signal, also known as the energy in the segment. If the amplitude levels are large then the segment is classified as voiced and if they are small then the segment is considered unvoiced. This determination requires a preconceived notion about the range of amplitude values and energy levels associated with the two types of sound. In order to help determine a classification for sounds that cannot be clearly classified based on an analysis of the amplitude, a second step is used to make the final distinction between voiced and unvoiced sounds. This step takes advantage of the fact that voiced speech segments have large amplitudes, unvoiced speech segments have high frequencies, and that the average values of both types of speech samples is close to zero. These three facts lead to the conclusion that the unvoiced speech waveform must cross the x-axis more often than the waveform of voiced speech. This can clearly be seen to be true in the case of Figure 3.3 and Figure 3.4. Thus, the determination of voiced and unvoiced speech signals is finalized by counting the number of times a waveform crosses the x-axis and then comparing that value to the normally range of values for most unvoiced and voiced sounds.

An additional factor that influences this classification is the surrounding segments. The classification of these neighbouring segments is taken into consideration because it is undesirable to have an unvoiced frame in the middle of a group of voiced frames or vice versa. It is important to realize that sound isn't always produced according to the LPC model. One example of this occurs when segments of voiced speech with a lot of background noise are sometimes interpreted as unvoiced segments. Another case of misinterpretation by the LPC model is with a group of sounds known as nasal sounds. During the production of these sounds the nose cavity destroys the concept of a linear tube since the tube now has a branch. This problem is often ignored in LPC but taken care of in other speech models which have the flexibility of higher bit rates. Not only is it possible to have sounds that are not produced according to the model, it is also possible to have sounds that are produced according to the model but that cannot be accurately classified as voiced or unvoiced. These sounds are a combination of the chaotic waveforms of unvoiced sounds and the periodic waveforms of voiced sounds. The LPC Model cannot accurately reproduce these sounds Examples of such sounds are "**th**is **z**oo" and "a**z**ure".

Another type of speech encoding called code excited linear prediction coding (CELP) handles the problem of sounds that are combinations of voiced and unvoiced by using a standard codebook which contains typical problematic signals. In the LPC model, only two different default signals are used to excite the filter for unvoiced and voiced signals in the decoder. In the CELP model, the encoder or synthesizer would compare a given waveform to the codebook and find the closest matching entry. This entry would be sent to the decoder which takes the entry code that is received and gets the corresponding entry from its codebook and uses this entry to excite the formant filter instead of one of the default signals used by LPC. CELP has a minimum of 4800 bits/second and can therefore afford to use a codebook. In LPC, which has half of the bit rate of CELP, the occasionally transmission of segments with problematic waveforms that will not be accurately reproduced by the decoder is considered an acceptable error.

3.4 Pitch Period Estimation

Determining if a segment is a voiced or unvoiced sound is not all of the information that is needed by the LPC decoder to accurately reproduce a speech signal**.** In order to produce an input signal for the LPC filter the decoder also needs another attribute of the current speech segment known as the pitch period. The period for any wave, including speech signals, can be defined as the time required for one wave cycle to completely pass a fixed position. For speech signals, the pitch period can be thought of as the period of the vocal cord vibration that occurs during the production of voiced speech. Therefore, the pitch period is only needed for the decoding of voiced segments and is not required for unvoiced segments since they are produced by turbulent air flow not vocal cord vibrations.

It is very computationally intensive to determine the pitch period for a given segment of speech. There are several different types of algorithms that could be used. One type of algorithm takes advantage of the fact that the autocorrelation of a period function $R_{xx}(K)$ will have a maximum when k is equivalent to the pitch period. These algorithms usually detect a maximum value by checking the autocorrelation value against a threshold value. One problem with algorithms that use autocorrelation is that the validity of their results is susceptible to interference as a result of other resonances in the vocal tract. When interference occurs the algorithm cannot guarantee accurate results. Another problem with autocorrelation algorithms occurs because voiced speech is not entirely periodic. This means that the maximum will be lower than it should be for a true periodic signal. LPC-10 does not use an algorithm with auto correlation; instead it uses an algorithm called average magnitude difference function (AMDF) which is defined as

$$\mathrm{AMDF(P)} = \frac{1}{\mathrm{N}} \sum_{i=k_0+1}^{k_0+N} |y_i + y_{i-p}|$$

Since the pitch period, P, for humans is limited; the AMDF is evaluated for a limited range of the possible pitch period values. Therefore, in LPC-10 there is an assumption that the pitch period is between 2.5 and 19.5 milliseconds. If the signal is sampled at a rate of 8000 samples/second then 20≤P ≤160.

For voiced segments we can consider the set of speech samples for the current segment, $\{y_n\}$, as a periodic sequence with period P_0. This means that samples that are Po apart should have similar values and that the AMDF function will have a minimum at P_0, that is when P is equal to the pitch period. An example of the AMDF function applied to the letter "e" from the word "test" can be seen in Figure 3.5. If this waveform is compared with the waveform for "e" before the AMDF function applied, in Figure 3.3, it can be seen that this function smooths out the waveform.

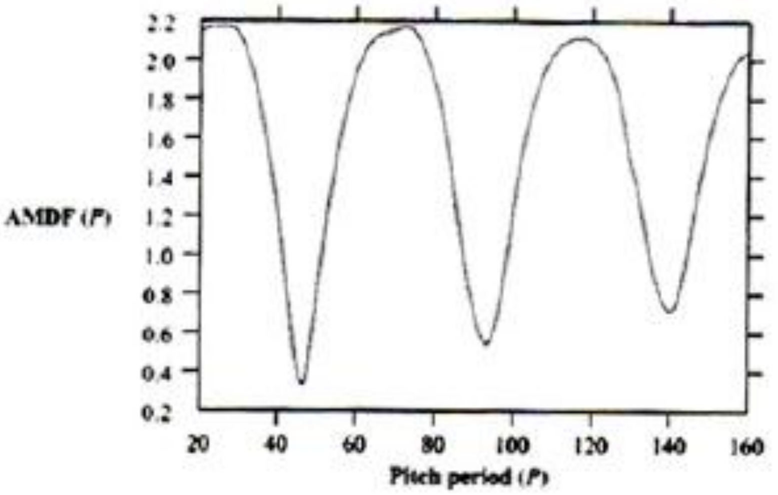


Fig. 3.5 AMDF function for voiced sound – Letter "e" in the word "test"

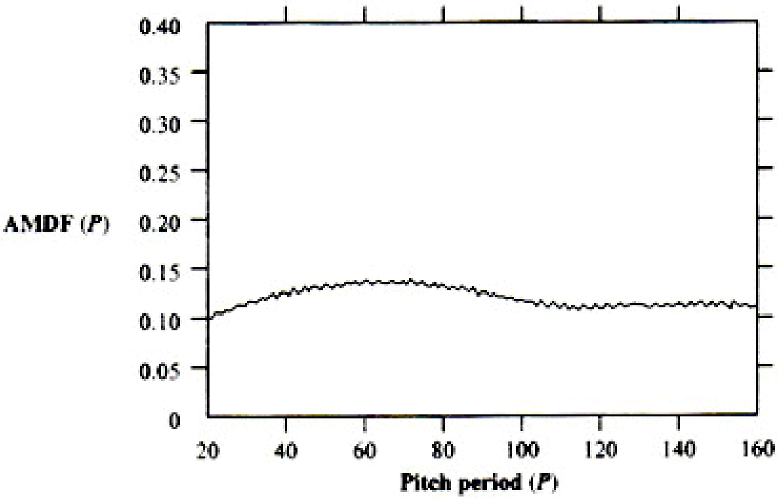


Fig. 3.6 AMDF function for unvoiced Sound – Letter "s" in the word "test"

3.5. Vocal Tract Filter

The filter that is used by the decoder to recreate the original input signal is created based on a set of coefficients. These coefficients are extracted from the original signal during encoding and are transmitted to the receiver for use in decoding. Each speech segment has different filter coefficients or parameters that it uses to recreate the original sound. Not only are the parameters themselves different from segment to segment, but the number of parameters differ from voiced to unvoiced segment. Voiced segments use 10 parameters to build the filter while unvoiced sounds use only 4 parameters. A filter with n parameters is referred to as an n^{th} order filter.

In order to find the filter coefficients that best match the current segment being analysed the encoder attempts to minimize the mean squared error. The mean squared error is expressed as:

$$e_n^2 = (y_n - \sum_{i=1}^{M} a_i y_{n-i} + G \in_n)^2$$

Where $\{y_n\}$ is the set of speech samples for the current segment and $\{a_i\}$ is the set of coefficients. In order to provide the most accurate coefficients,$\{a_i\}$ is chosen to minimize the average value of e_n^2 for all samples in the segment.

The first step in minimizing the average mean squared error is to take the derivative.

$$\frac{\delta}{\delta a_j} E[(y_n - \sum_{i=1}^{M} a_i y_{n-i} + G \in_n)^2] = 0$$

$$\rightarrow \ -2E[(y_n - \sum_{i=1}^{M} a_i y_{n-i} + G \in_n) y_{n-j}] = 0$$

$$\rightarrow \ \sum_{i=1}^{M} a_i E[y_{n-i}\, y_{n-j}] = E[y_n y_{n-j}]$$

(Use fact that $E[y_n y_{n-j}]$=0 if j $\neq$ 0)

Taking the derivative produces a set of M equations. In order to solve for the filter coefficients E $[y_{n-i} y_{n-j}]$ has to be estimate. There are two approaches that can be used for this estimation: autocorrelation and auto covariance. Although there are versions of LPC that use both approaches.

Autocorrelation requires that several initial assumptions be made about the set or sequence of speech samples, $\{y_n\}$, in the current segment. First, it requires that $\{y_n\}$ be stationary and second, it requires that the $\{y_n\}$ sequence is zero outside of the current segment. In autocorrelation, each E $[y_{n-i}y_{n-j}]$ is converted into an autocorrelation function of the form$R_{yy}(|i-j|)$). The estimation of an autocorrelation function $R_{yy}(K)$ can be expressed as:

$$R_{yy}(k) = \sum_{n=n_o+1+k}^{n_o+N} y_n y_{n-k}$$

Using $R_{yy}(K)$, the M equations that were acquired from taking the derivative of the mean squared error can be written in matrix form RA = P where A contains the filter coefficients.

$$\mathrm{R} = \begin{bmatrix} R_{yy}(0) & R_{yy}(1) & \dots & R_{yy}(M-1) \\ R_{yy}(1) & R_{yy}(0) & \dots & R_{yy}(M-2) \\ \dots & \dots & \dots & \dots \\ R_{yy}(M-1) & R_{yy}(M-2) & \dots & R_{yy}(0) \end{bmatrix}$$

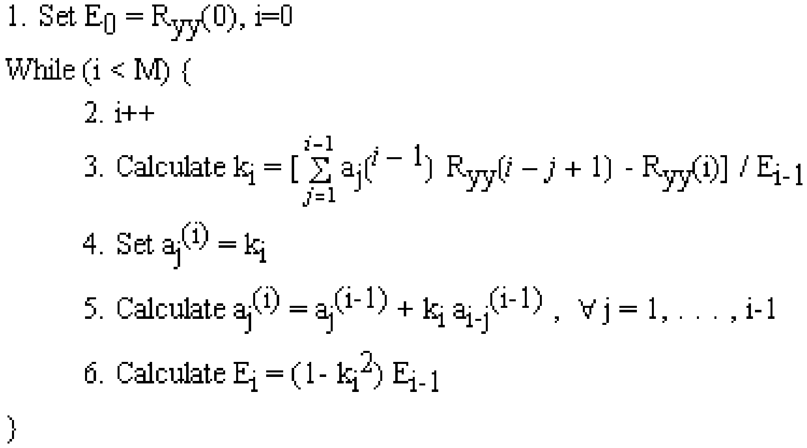
$$\mathrm{A} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_M \end{bmatrix} \qquad \mathrm{P} = \begin{bmatrix} R_{yy}(1) \\ R_{yy}(2) \\ \vdots \\ R_{yy}(M) \end{bmatrix}$$

In order to determine the contents of A, the filter coefficients, the equation A = $R^{-1}P$ must be solved. This equation cannot be solved without first computing R^{-1}. This is an easy computation if one notices that R is symmetric and more importantly all diagonals consist of the same element. This type of matrix is called a Toeplitz matrix and can be easily inverted.

The Levinson-Durbin (L-D) Algorithm is a recursive algorithm that is considered very computationally efficient since it takes advantage of the properties of R when determining the filter coefficients.

3.6 Levinson-Durbin (L-D) Algorithm

The Levinson-Durbin (L-D) Algorithm is a recursive algorithm that is considered very computationally efficient since it takes advantage of the properties of R when determining the filter coefficients.



```
1. Set $E_0 = R_{yy}(0)$, $i=0$
While ($i < M$) {
        2. $i$++
        3. Calculate $k_i = [\sum_{j=1}^{i-1} a_j^{(i-1)} R_{yy}(i - j + 1) - R_{yy}(i)] / E_{i-1}$
        4. Set $a_j^{(i)} = k_i$
        5. Calculate $a_j^{(i)} = a_j^{(i-1)} + k_i a_{i-j}^{(i-1)}$ ,  $\forall j = 1, . . . , i-1$
        6. Calculate $E_i = (1 - k_i^2) E_{i-1}$
}
```

Fig. 3.7 Levinson-Durbin (L-D) Algorithm for solving Toeplitz Matrices

During the process of computing the filter coefficients $\{a_i\}$ a set of coefficients, $\{k_i\}$, called reflection coefficients or partial correlation coefficients (PARCOR) are generated. These coefficients are used to solve potential problems in transmitting the filter coefficients. The quantization of the filter coefficients for transmission can create a major problem since errors in the filter coefficients can lead to instability in the vocal tract filter and create an inaccurate output signal. This potential problem is averted by quantizing and transmitting the reflection coefficients that are generated by the Levinson-Durbin algorithm. These coefficients can be used to rebuild the set of filter coefficients $\{a_i\}$ and can guarantee a stable filter if their magnitude is strictly less than one.

3.6.1. Transmitting the Parameters

In an uncompressed form, speech is usually transmitted at 64,000 bits/second using 8 bits/sample and a rate of 8 kHz for sampling. LPC reduces this rate to 2,400 bits/second by breaking the speech into segments and then sending the voiced/unvoiced information, the pitch period, and the coefficients for the filter that represents the vocal tract for each segment.

The input signal used by the filter on the receiver end is determined by the classification of the speech segment as voiced or unvoiced and by the pitch period of the segment. The encoder sends a single bit to tell if the current segment is voiced or unvoiced. The pitch period is quantized using a log-companded quantizer to one of 60 possible values. 6 bits are required to represent the pitch period.

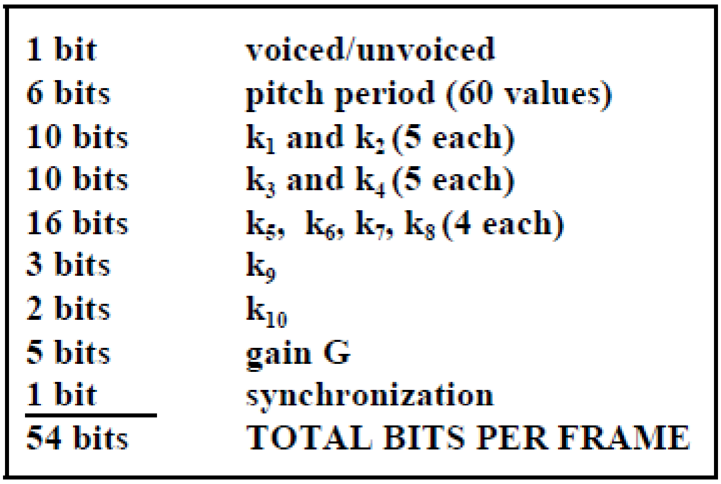
If the segment contains voiced speech than an 10^{th} order filter is used. This means that 11 values are needed: 10 reflection coefficients and the gain. If the segment contains unvoiced speech than an 4^{th} order filter is used. This means that 5 values are needed: 4 reflection coefficients and the gain. The reflection coefficients are denoting k_n where 1 ≤ n ≤10 for voiced speech filters and 1 ≤ n≤4 for unvoiced filters.

The only problem with transmitting the vocal tract filter is that it is especially sensitive to errors in reflection coefficients that have a magnitude close to one. The first few reflection coefficients, k_1 and k_2, are the most likely coefficients to have magnitudes around one. To try and eliminate this problem, LPC-10 uses non uniform quantization for k_1 and k_2. First, each coefficient is used to generate a new coefficient, g_i, of the form

$$g_i = \frac{1+k_i}{1-k_i}$$

These new coefficients g_1 and g_2 , are then quantized using a 5-bit uniform quantizer.

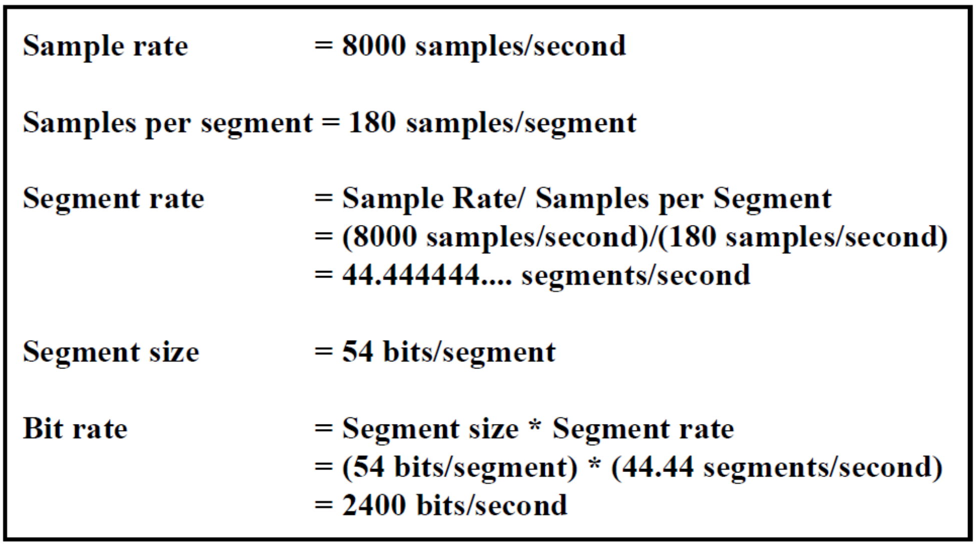
All of the rest of the reflection coefficients are quantized using uniform quantizers. k_3 and k_4 are quantized using 5-bit uniform quantization. For voiced segments k_5 up to k_8 are quantized using 4-bit uniform quantizers, while k_9 uses a 3-bit quantizer and k_{10} uses a 2-bit uniform quantizer. For unvoiced segments the bits used in voiced segments to represent the reflection coefficients, k_5 through k_{10}, are used for error protection. This means that unvoiced segments don’t omit the bits needed to represent k_5 up to k_{10} and therefore use the same amount of space as voiced segments. This also means that the bit rate doesn’t decrease below 2,400 bits/second if a lot of unvoiced segments are sent. Variation of bit rate is not good in the transmission of speech since most speech is transmitted over shared lines where it is important to know how much of the line will be needed. Once the reflection coefficients have been quantized the gain, G, is the only thing not yet quantized. The gain is calculated using the root mean squared (rms) value of the current segment. The gain is quantized using a 5-bit log-companded quantizer.



1 bit	voiced/unvoiced
6 bits	pitch period (60 values)
10 bits	k_1 and k_2 (5 each)
10 bits	k_3 and k_4 (5 each)
16 bits	k_5, k_6, k_7, k_8 (4 each)
3 bits	k_9
2 bits	k_{10}
5 bits	gain G
1 bit	synchronization
54 bits	TOTAL BITS PER FRAME

Fig. 3.8 Total Bits in each Speech Segment

The total number of bits required for each segment or frame is 54bits which is explained in fig 3.8.The input speech is sampled at a rate of 8000 samples per second and that the 8000 samples in each second of speech signal are broken into 180 sample segments. This means that there are approximately 44.4 frames or segments per second and therefore the bit rate is 2400 bits/second as shown in fig 3.9.



Sample rate = 8000 samples/second

Samples per segment = 180 samples/segment

Segment rate = Sample Rate/ Samples per Segment
= (8000 samples/second)/(180 samples/second)
= 44.444444.... segments/second

Segment size = 54 bits/segment

Bit rate = Segment size * Segment rate
= (54 bits/segment) * (44.44 segments/second)
= 2400 bits/second

Fig. 3.9 Verification for Bit Rate of LPC Speech Segments

CHAPTER 4

LINEAR PREDICTION SYNTHESIS

CHAPTER 4

LINEAR PREDICTION SYNTHESIS

4.1 Synthesis of speech signals

Speech synthesis is the artificial production of human speech. A computer system used for this purpose is called a speech synthesizer, and can be implemented in software or hardware products. A text-to-speech (TTS) system converts normal language text into speech; other systems render symbolic linguistic representations like phonetic transcriptions into speech.

Synthesized speech can be created by concatenating pieces of recorded speech that are stored in a database. Systems differ in the size of the stored speech units; a system that stores phones or diphones provides the largest output range, but may lack clarity. For specific usage domains, the storage of entire words or sentences allows for high-quality output. Alternatively, a synthesizer can incorporate a model of the vocal tract and other human voice characteristics to create a completely "synthetic" voice output.

The quality of a speech synthesizer is judged by its similarity to the human voice and by its ability to be understood. An intelligible text-to-speech program allows people with visual impairments or reading disabilities to listen to written works on a home computer. Many computer operating systems have included speech synthesizers since the early 1990s.

A text-to-speech system (or "engine") is composed of two parts a front-end and a back-end. The front-end has two major tasks. First, it converts raw text containing symbols like numbers and abbreviations into the equivalent of written-out words. This process is often called text normalization, pre-processing, or tokenization. The front-end then assigns phonetic transcriptions to each word, and divides and marks the text into prosodic units, like phrases, clauses, and sentences. The process of assigning phonetic transcriptions to words is called text-to-phoneme or grapheme-to-phoneme conversion. Phonetic transcriptions and prosody information together make up the symbolic linguistic representation that is output by the front-end. The back-end—often referred to as the synthesizer—then converts the symbolic linguistic representation into sound. In certain systems, this part includes the computation of the target prosody (pitch contour, phoneme durations), which is then imposed on the output speech.

4.2 LPC Synthesis/Decoding

The process of decoding a sequence of speech segments is the reverse of the encoding process. Each segment is decoded individually and the sequence of reproduced sound segments is joined together to represent the entire input speech signal. The decoding or synthesis of a speech segment is based on the 54 bits of information that are transmitted from the encoder.

The speech signal is declared voiced or unvoiced based on the voiced/unvoiced determination bit. The decoder needs to know what type of signal the segment contains in order to determine what type of excitement signal will be given to the LPC filter. Unlike other speech compression algorithms like CELP which have a codebook of possible excitement signals, LPC has only two possible signals. For voiced segments a pulse is used as the excitement signal. This pulse consists of 40 samples and is locally stored by the decoder. A pulse is defined as "...an isolated disturbance, that travels through an otherwise undisturbed medium" [10]. For unvoiced segments white noise produced by a pseudorandom number generator is used as the input for the filter.

The pitch period for voiced segments is then used to determine whether the 40 sample pulse needs to be truncated or extended. If the pulse needs to be extended it is padded with zeros since the definition of a pulse said that it travels through an undisturbed medium. This combination of voice/unvoiced determination and pitch period are the only things that are need to produce the excitement signal.

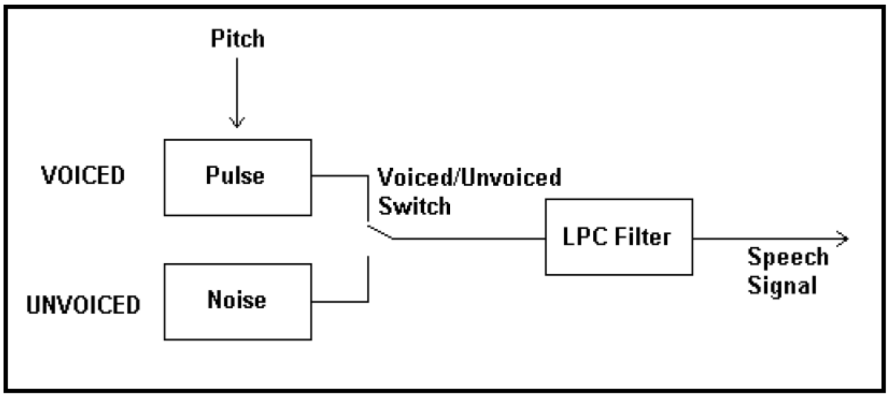


Fig 4.1 LPC Decoder

Each segment of speech has a different LPC filter that is eventually produced using the reflection coefficients and the gain that are received from the encoder. 10 reflection coefficients are used for voiced segment filters and 4 reflection coefficients are used for unvoiced segments. These reflection coefficients are used to generate the vocal tract coefficients or parameters which are used to create the filter.

The final step of decoding a segment of speech is to pass the excitement signal through the filter to produce the synthesized speech signal. Figure 4.1 shows a diagram of the LPC decoder.

The complete block diagram of an LPC speech analysis–synthesis system is given in Fig. 4.2.

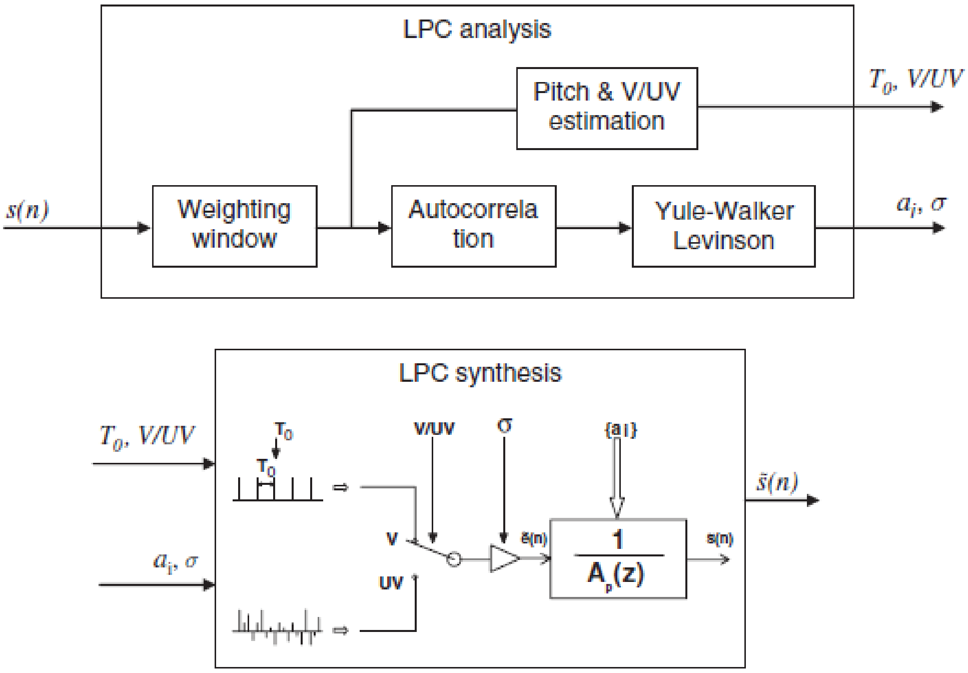


Fig. 4.2 A linear predictive speech analysis–synthesis system

4.3 Linear predictive coders

The LPC analysis–synthesis system, which has been described above, is not exactly the one embedded in cell phones. It is, however, implemented in the so-called NATO LPC10 standard (NATO, 1984), which was used for satellite transmission of speech communications until 1996. This norm makes it possible to encode speech with a bit rate as low as 2,400 bits/s (frames are 22.5 ms long, and each frame is coded with 54 bits: 7 bits for pitch and V/UV decision, 5 bits for the gain, and 42 bits for the prediction coefficients4). In practice, prediction coefficients are actually not used as such; the related reflection coefficients or log area ratios are preferred, since they have better quantization properties. Quantization of prediction coefficients can result in unstable filters.

The number of bits in LPC10 was chosen such that it does not bring audible artifacts to the LPC speech. The example LPC speech produced is therefore a realistic example of typical LPC10 speech. Clearly this speech coder suffers from the limitations of the poor (and binary!) excitation model. Voiced fricatives, for instance, cannot be adequately modelled since they exhibit voiced and unvoiced features simultaneously. Moreover, the LPC10 coder is very sensitive to the efficiency of its voiced/unvoiced detection and F_0 estimation algorithms. Female voices, whose higher F_0 frequency sometimes results in a second harmonic at the center of the first formant, often lead to F_0 errors (the second harmonic being mistaken for F_0).

One way of enhancing the quality of LPC speech is obviously to reduce the constraints on the LPC excitation so as to allow for a better modelling of the prediction residual e(n) by the excitation $\tilde{e}(n)$. As a matter of fact, passing this residual through the synthesis filter 1/A(z) produces the original speech.

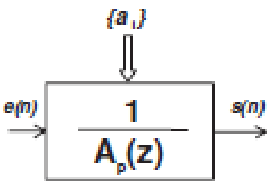


Fig. 4.3 passing the prediction residual through the synthesis filter produces the original speech signal

The multi pulse excited (MPE; Atal and Remde 1982) was an important step in this direction, as it was the first approach to implement an analysis by- synthesis process (i.e., a closed loop) for the estimation of the excitation features. The MPE excitation is characterized by the positions and amplitudes of a limited number of pulses per frame (typically 10 pulses per 10 ms frame; Fig. 4.4). Pitch estimation and voiced/unvoiced decision are no longer required. Pulse positions and amplitudes are chosen iteratively (Fig. 4.5) so as to minimize the energy of the modelling error (the difference between the original speech and the synthetic speech). The error is filtered by a perceptual filter before its energy is computed:

$$P(z) = \frac{A(z)}{A(z/\gamma)}$$

The role of this filter, whose frequency response can be set to any intermediate between all pass response (γ =1) and the response of the inverse filter (γ = 0), is to reduce the contributions of the formants to the estimation of the error. The value of γ is typically set to 0.8.

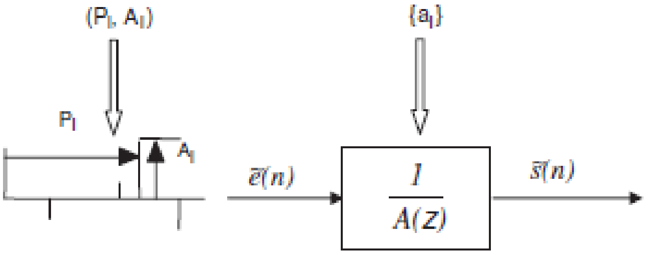


Fig. 4.4 the MPE decoder

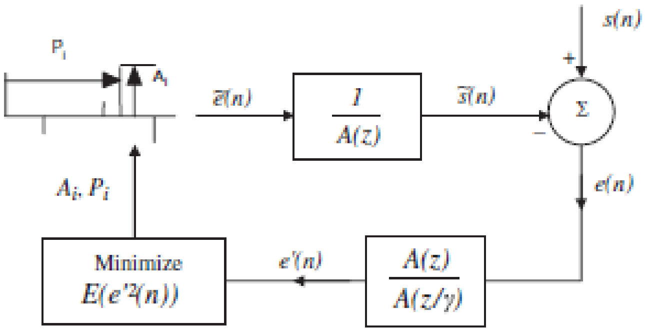


Fig. 4.5 Estimation of the MPE excitation by an analysis-by-synthesis loop in the MPE encoder

The code-excited linear prediction (CELP) coder (Schroeder and Atal, 1985) further extended the idea of analysis-by-synthesis speech coding by using the concept of vector quantization (VQ) for the excitation sequence. In this approach, the encoder selects one excitation sequence from a predefined stochastic codebook of possible sequences (Fig. 4.6) and sends only the index of the selected sequence to the decoder, which has a similar codebook. Although the lowest quantization rate for scalar quantization is 1 bit per sample, VQ allows fractional bit rates. For example, quantizing two samples simultaneously using a 1-bit codebook will result in 0.5 bits per sample. More typical values are a 10-bit codebook with codebook vectors of dimension 40, resulting in 0.25 bits per sample. Given the very high variability of speech frames, however (due to changes in glottal excitation and vocal tract), vector-quantized speech frames would be possible only with a very large codebook. The great idea of CELP is precisely to perform VQ on LP residual sequences; the LP residual has a flat spectral envelope, which makes it easier to produce a small but somehow exhaustive codebook of LP residual sequences. CELP can thus be seen as an adaptive vector quantization scheme of speech frames (adaptation being performed by the synthesis filter).

CELP additionally takes advantage of the periodicity of voiced sounds to further improve predictor efficiency. A so-called long-term predictor filter is cascaded with the synthesis filter, which enhances the efficiency of the codebook. The simplest long-term predictor consists of a simple variable delay with adjustable gain (Fig. 4.7).

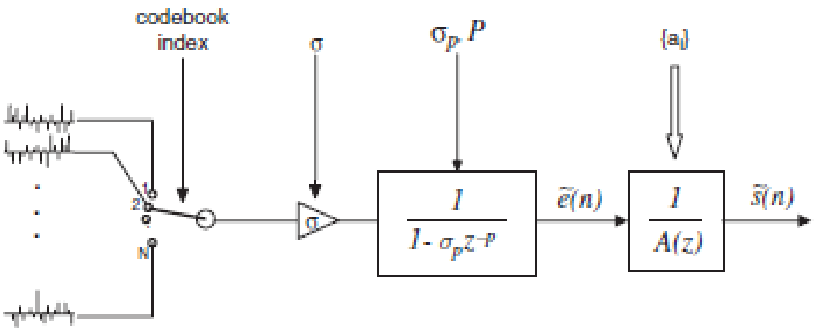


Fig. 4.6 the CELP decoder

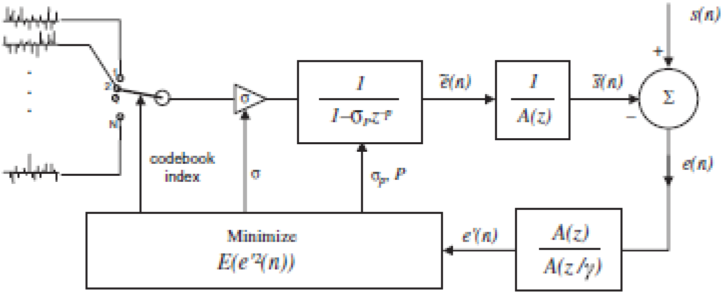


Fig. 4.7 Estimation of the CELP excitation by an analysis-by-synthesis loop in the CELP encoder

Various coders have been developed after MPE and CELP using the same analysis-by-synthesis principle with the goal of enhancing CELP quality while further reducing bit rate, among which are the mixed excitation linear prediction (MELP; McCree and Barnwell, 1995) and the harmonic and vector excitation coding (HVXC; Matsumoto et al. 1997). In 1996, LPC-10 was replaced by MELP to be the United States Federal Standard for coding at 2.4 kbps.

From 1992 to 1996, GSM (global system for mobile communication) phones embedded a particular form of MPE, the RPE-LPC (regular pulse excited; Kroon et al. 1986) coder, with additional constraints on the positions of the pulses: the RPE pulses were evenly spaced (but their amplitude, as well as the position of the first pulse, is left open). Speech is divided into 20 ms frames, each of which is encoded as 260 bits, giving a total bit rate of 13 kbps. In 1996, this so-called full-rate (FR) codec was replaced by the enhanced full-rate (EFR) codec, implementing a variant of CELP termed as algebraic-CELP (ACELP, Salami et al. 1998). The ACELP codebook structure allows efficient searching of the optimal codebook index thereby eliminating one of the main drawbacks of CELP which is its complexity. The EFR coder operates at 11.2 kbps and produces better speech quality than the FR coder at 13 kb/s. A variant of the ACELP coder has been standardized by ITU-T as G.729 for operation at a bit rate of 8 kbps. Newer generations of coders that are used in cell phones are all based on the CELP principle and can operate at bit rates as low as 4.75 – 11.2 kbps.

CHAPTER 5

SIMULATION RESULTS AND DISCUSSION

CHAPTER 5

SIMULATION RESULTS AND DISCUSSION

5.1 Examining a speech file

Let us load file “speech.wav,” listen to it, and plot its samples (Fig. 5.1). This file contains the sentence “Paint the circuits” sampled at 8 kHz, with 16 bits.

```
speech=wavread('C:\Users \Desktop\sp20.wav');
figure(1);
plot(speech)
xlabel('time');
ylabel('amplitude');
sound(speech,8000);
```

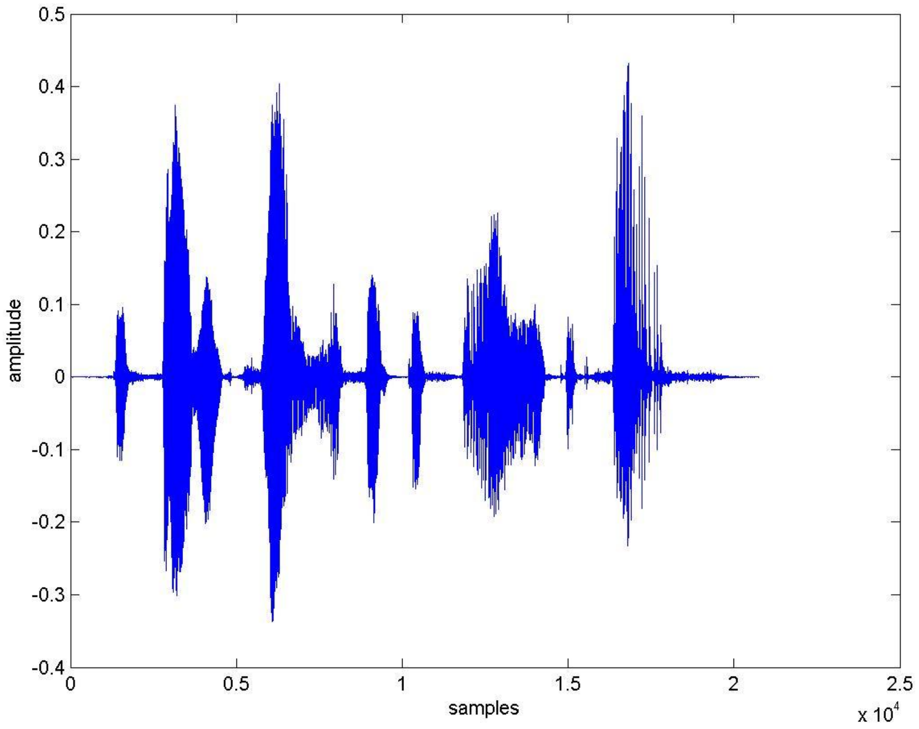


Fig. 5.1 Input speech waveform

The file is about 1.1 s long (9,000 samples). One can easily spot the position of the four vowels appearing in this plot, since vowels usually have higher amplitude than other sounds.

As such, however, the speech waveform is not "readable," even by an expert phonetician. Its information content is hidden. In order to reveal it to the eyes, let us plot a spectrogram of the signal (Fig. 5.2). We then choose a wideband spectrogram7 by imposing the length of each frame to be approximately 5 ms long (40 samples) and a hamming weighting window.

```
figure(2);
specgram(speech,512,8000,hamming(40));
```

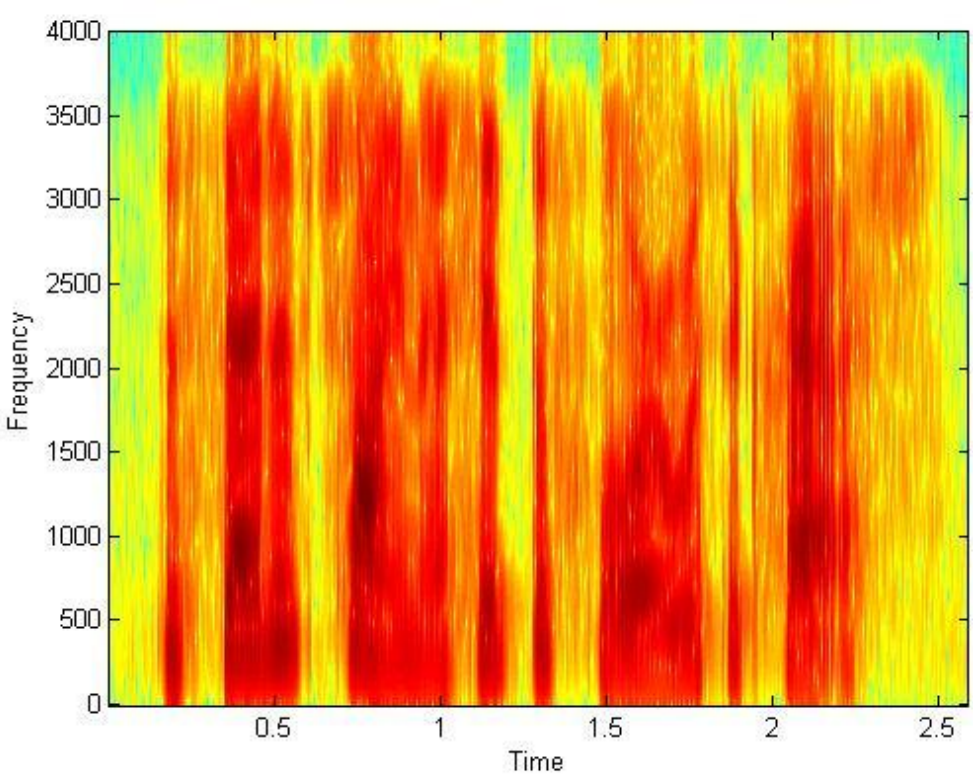


Fig. 5.2 Input speech specgram

In this plot, pitch periods appear as vertical lines. As a matter of fact, since the length of analysis frames is very small, some frames fall on the peaks (resp., on the valleys) of pitch periods and thus appear as a darker (resp., lighter) vertical lines.

In contrast, formants (resonant frequencies of the vocal tract) appear as dark (and rather wide) horizontal traces. Although their frequency is not easy to measure with precision, experts looking at such a spectrogram can actually often read it (i.e., guess the corresponding words). This clearly shows that formants are a good indicator of the underlying speech sounds.

5.2 Linear prediction synthesis of 30 ms of voiced speech

Let us extract a 30-ms frame from a voiced part (i.e., 240 samples) of the speech file and plot its samples (Fig. 5.3).

```
input_frame=speech(3000:4739);
figure(3);
plot(input_frame)
xlabel('time');
ylabel('amplitude');
```

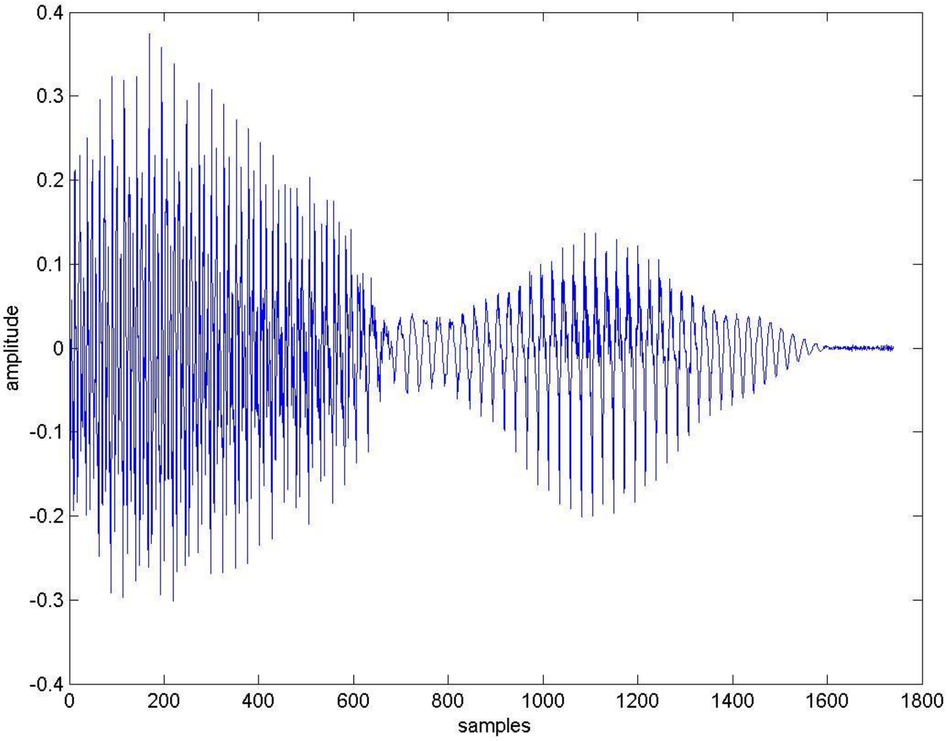


Fig. 5.3 Waveform of 30-ms-long voiced speech frame taken from a vowel

As expected this sound is approximately periodic (period=65 samples, i.e., 80 ms; fundamental frequency = 125 Hz). Note, though, that this is only apparent; in practice, no sequence of samples can be found more than once in the frame.

Now let us see the spectral content of this speech frame (Fig. 5.4) by plotting its periodogram on 512 points (using a normalized frequency axis; remember π corresponds to $F_S/2$, i.e., to 4,000 Hz here).

```
figure(4);
periodogram(input_frame,[],512);
```

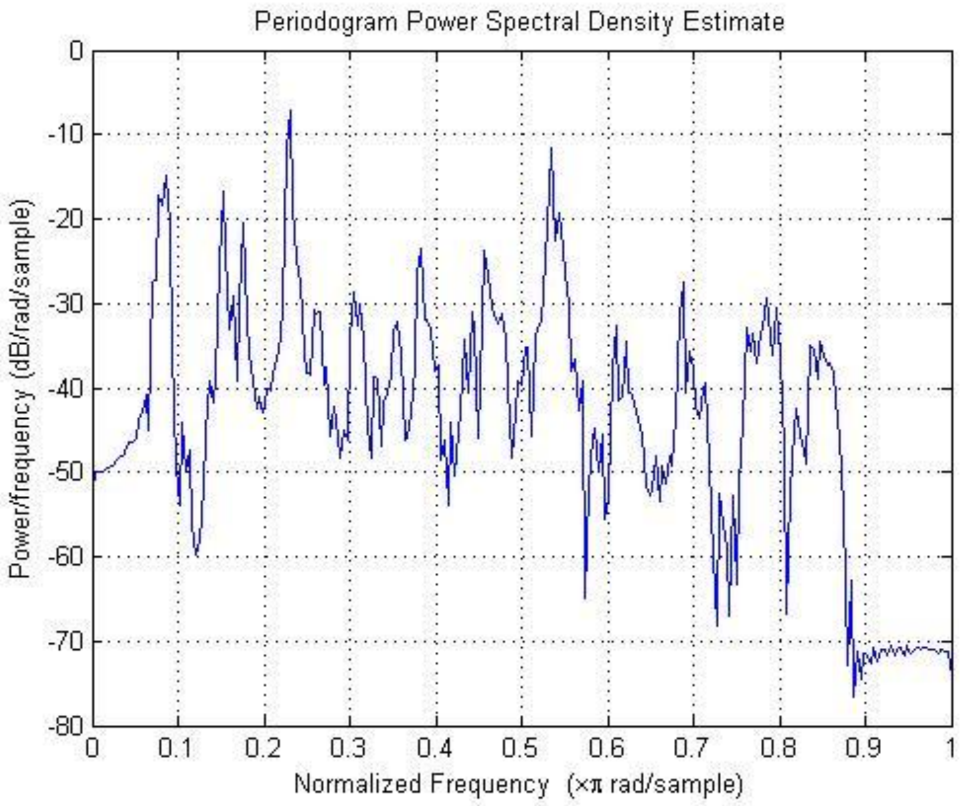


Fig. 5.4 Periodogram of 30-ms-long voiced speech frame taken from a vowel

The fundamental frequency appears again at around 125 Hz. One can also roughly estimate the position of formants (peaks in the spectral envelope) at ± 300, 1,400, and 2,700 Hz.

Let us now fit an LP model of order 10 to our voiced frame. We obtain the prediction coefficients (ai) and the variance of the residual signal(sigma_square).

```
[ai,sigma_square]=lpc(input_frame,10);
sigma=sqrt(sigma_square);
```

The estimation parameter inside LPC is called the Levinson–Durbin algorithm. It chooses the coefficients of an FIR filter *A(z)* so that when passing the input frame into *A(z)*, the output, termed as the prediction residual, has minimum energy. It can be shown that this leads to a filter which has anti-resonances wherever the input frame has a formant. For this reason, the *A(z)* filter is termed as the “inverse” filter. Let us plot its frequency response (on 512 points) and superimpose it to that of the “synthesis” filter *1/A(z)* (Fig. 5.5).

```
[HI,WI]=freqz(ai,1,512);
[H,W]=freqz(1,ai,512);
figure(5);
plot(W,20*log10(abs(H)),'-',WI,20*log10(abs(HI)),'--');
xlabel('normalized frequency');
ylabel('magnitude');
```

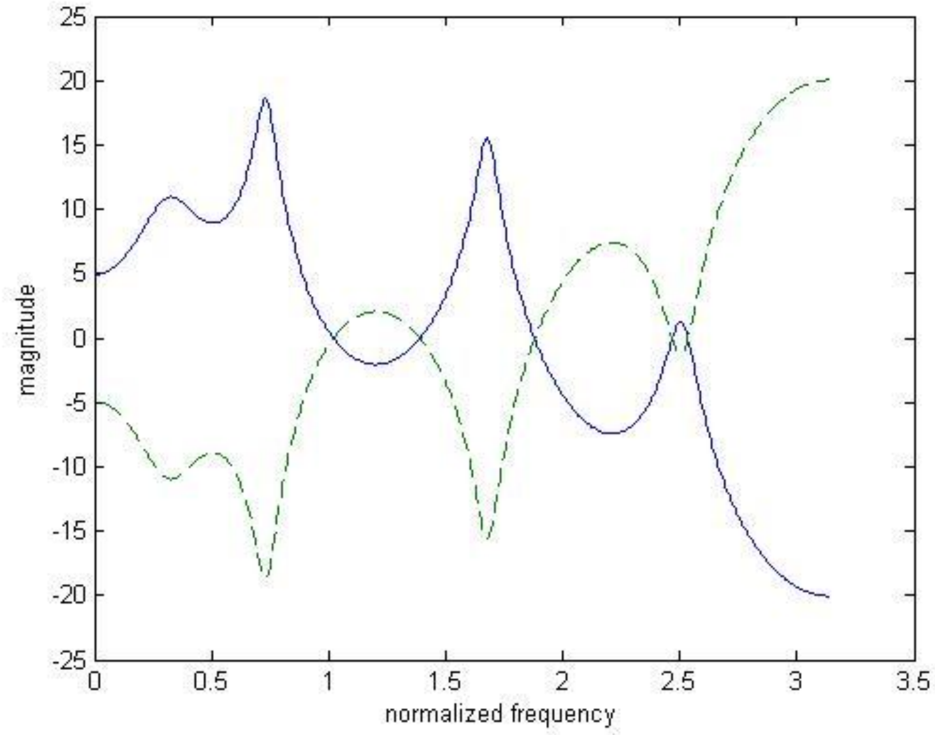


Fig. 5.5 Frequency responses of the inverse and synthesis filters

In other words, the frequency response of the filter *1/A(z)* matches the spectral amplitude envelope of the frame. Let us superimpose this frequency response to the periodogram of the vowel (Fig. 5.6).

```
figure(6);
periodogram(input_frame,[],512,2)
hold on;
plot(W/pi,20*log10(sigma*abs(H)));
xlabel('frequency');
ylabel('power density');
hold off;
```

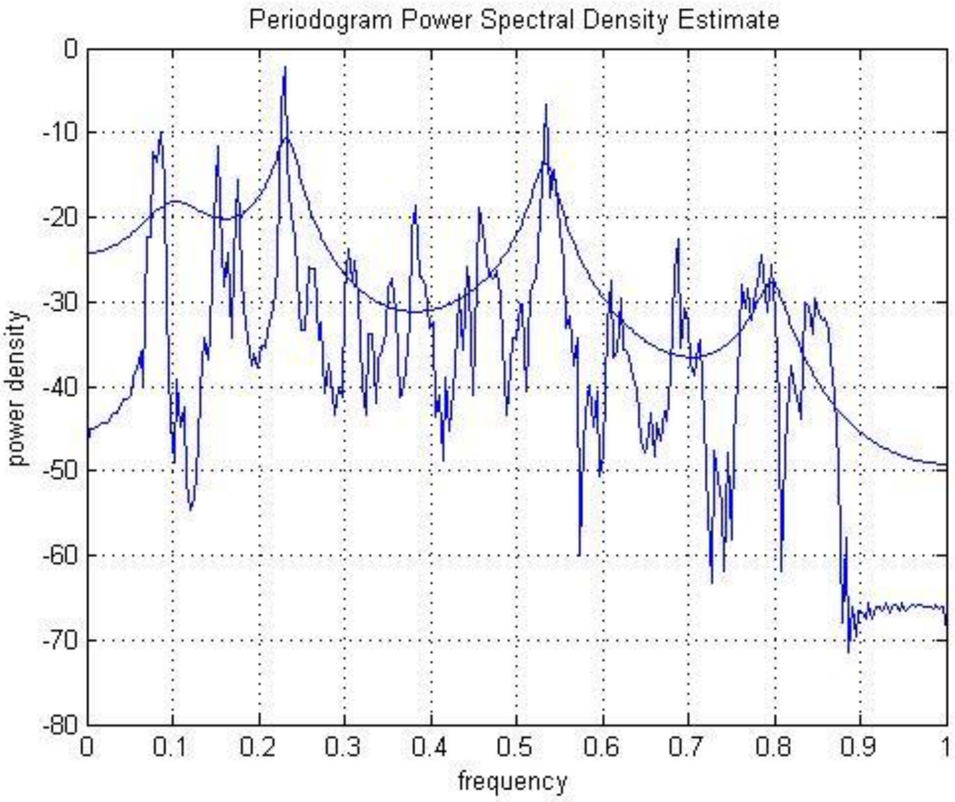


Fig. 5.6 Frequency response of the synthesis filter superimposed with the periodogram of the frame

In other words, the LPC fit has automatically adjusted the poles of the synthesis filter close to the unit circle at angular positions chosen to imitate formant resonances (Fig. 5.7).

```
figure(7);
zplane(1,ai);
```

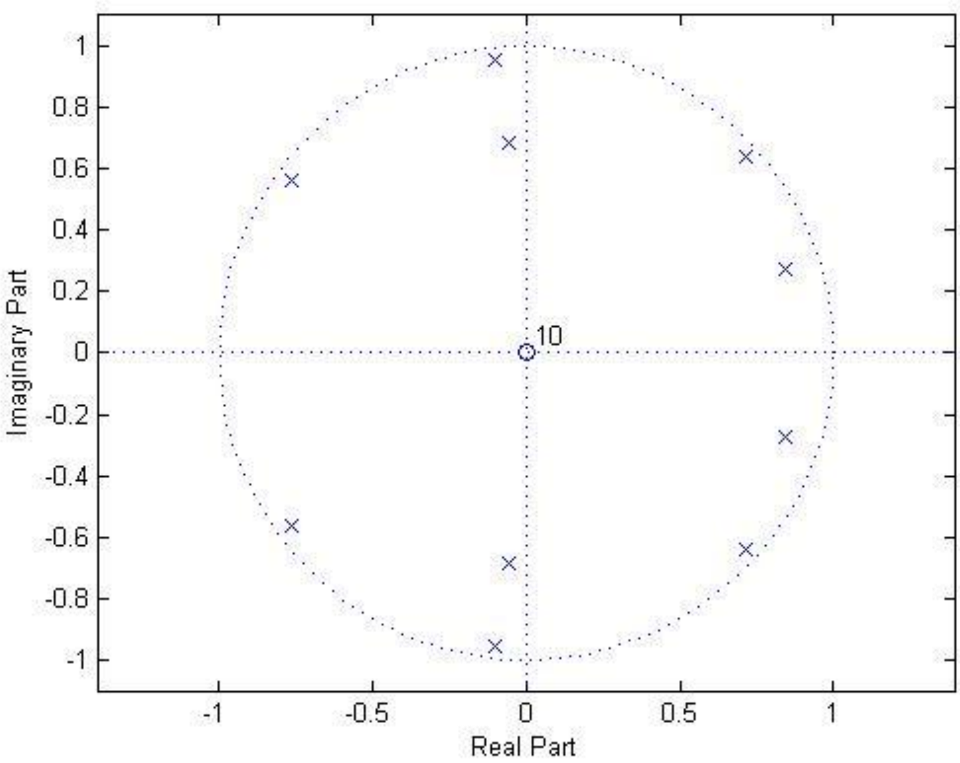


Fig. 5.7 poles and zeros of the filter

If we apply the inverse of this filter to the input frame, we obtain the prediction residual (Fig. 5.8).

```
LP_residual=filter(ai,1,input_frame);
figure(8);
plot(LP_residual)
xlabel('time');
ylabel('amplitude');
figure(9);
periodogram(LP_residual,[],512);
```

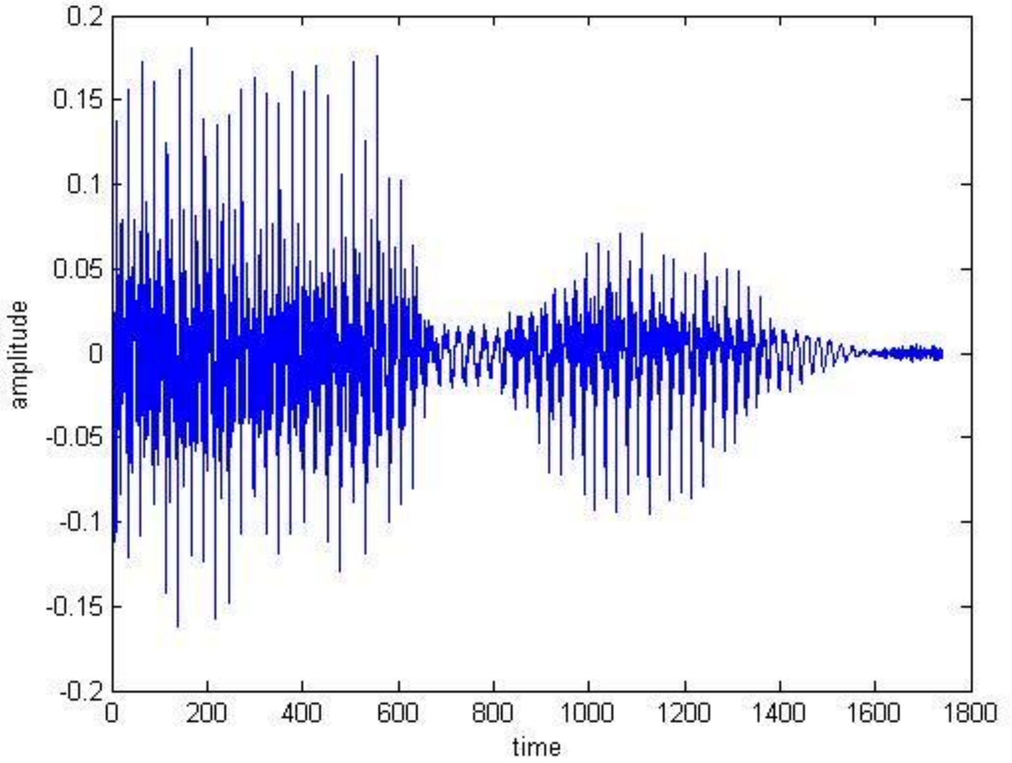


Fig. 5.8 Waveform of the prediction residual

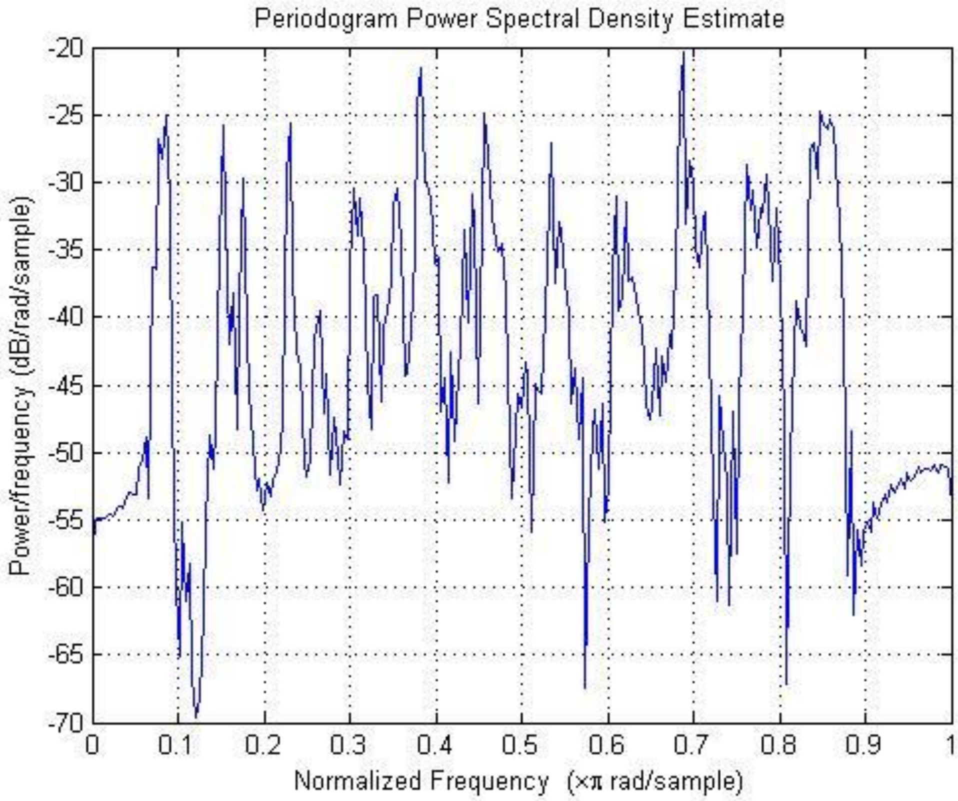


Fig. 5.9 Periodogram of the prediction residual

Let us compare the spectrum of this residual to the original spectrum. The new spectrum is approximately flat; its fine spectral details, however, are the same as those of the analysis frame. In particular, its pitch and harmonics are preserved.

For obvious reasons, applying the synthesis filter to this prediction residual results in the analysis frame itself (since the synthesis filter is the inverse of the inverse filter).

```
figure(10);
output_frame=filter(1, ai,LP_residual);
plot(output_frame);
xlabel('time');
ylabel('amplitude');
```

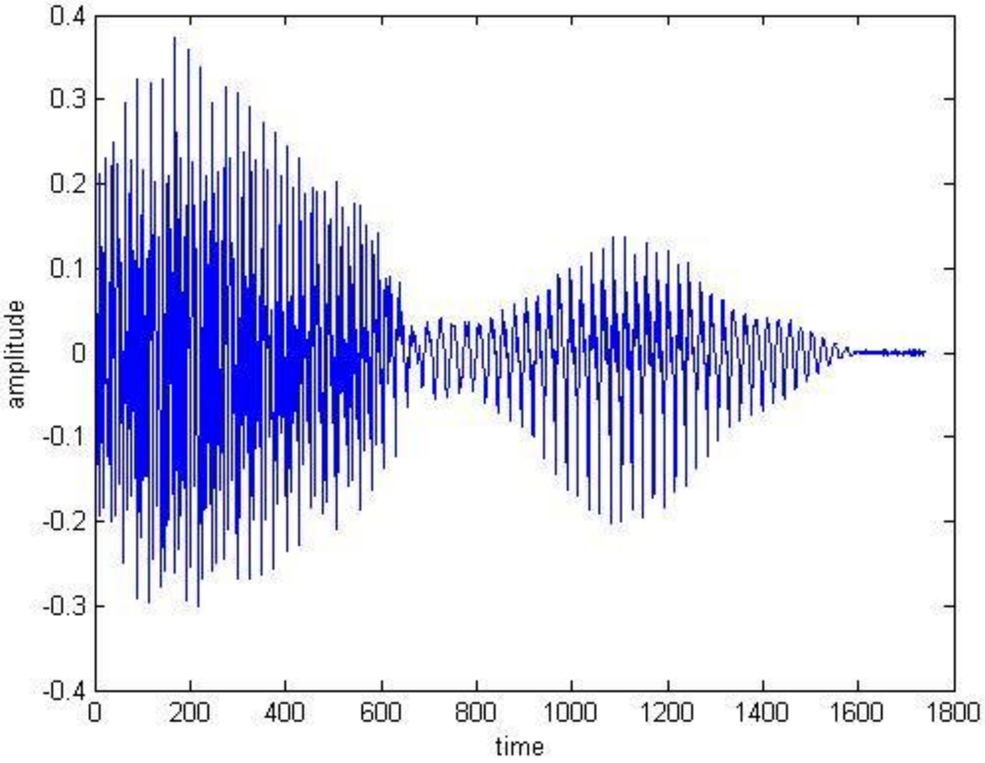


Fig. 5.10 Synthesis filter response

The LPC model actually models the prediction residual of voiced speech as an impulse train with adjustable pitch period and amplitude. For the speech frame considered, for instance, the LPC ideal excitation is a sequence of pulses separated by 64 zeros (so as to impose a period of 65samples; Fig. 5.11). Note we multiply the excitation by some gain so that its variance matches that of the residual signal.

```
excitation = [1;zeros(64,1);1;zeros(64,1);1;zeros(64,1);1;zeros(44,1)];
gain=sigma/sqrt(1/65);
figure(11);
plot(gain*excitation);
xlabel('time');
ylabel('amplitude');
figure(12);
periodogram(gain*excitation,[],512);
```

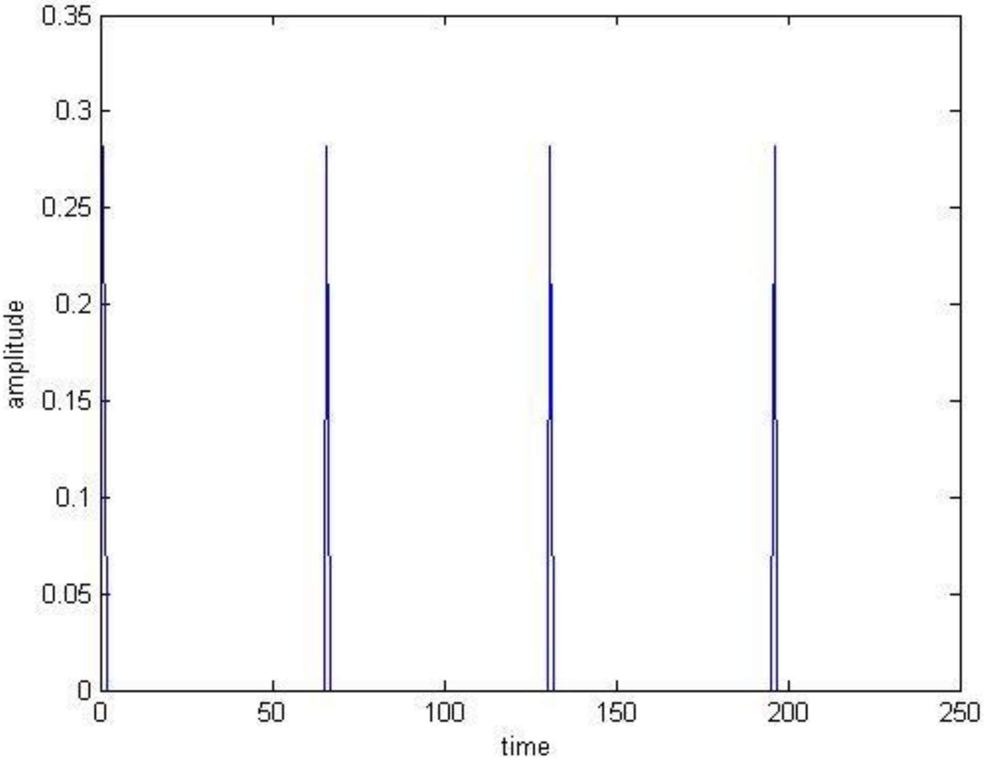


Fig. 5.11 Waveform of LPC excitation

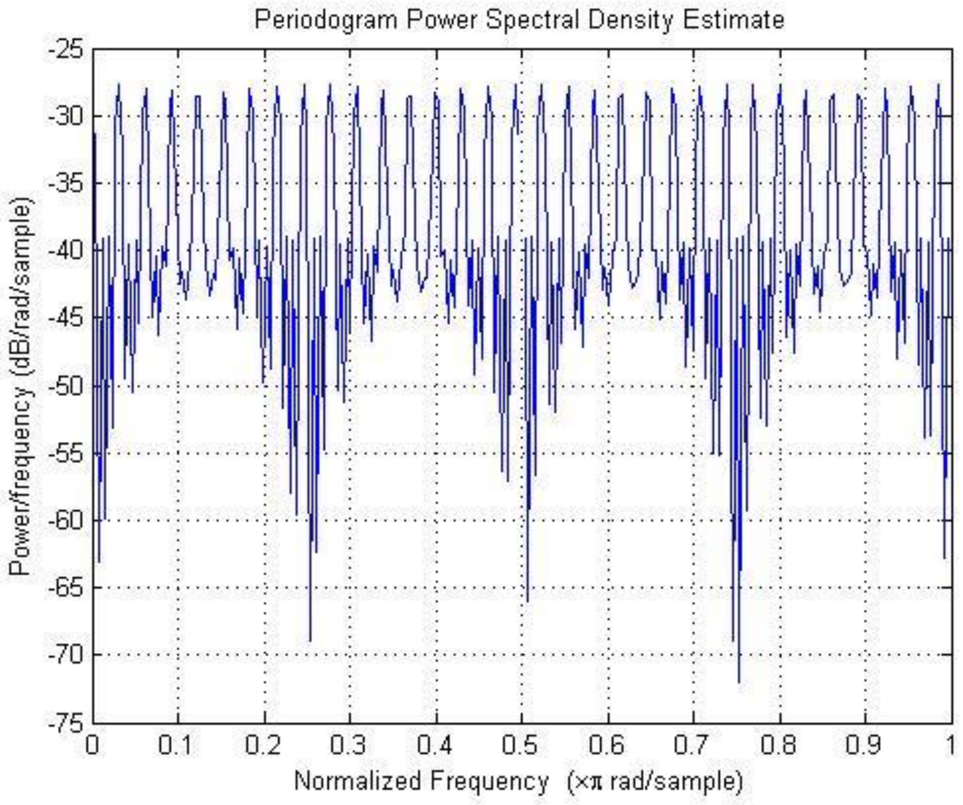


Fig. 5.12 periodogram of LPC excitation

Clearly, as far as the waveform is concerned, the LPC excitation is far from similar to the prediction residual. Its spectrum (Fig. 5.12), however, has the same broad features as that of the residual: flat envelope and harmonic content corresponding to F_0 . The main difference is that theexcitation spectrum is “over-harmonic” compared to the residual spectrum. Let us now use the synthesis filter to produce an artificial “e.”

```
synt_frame=filter(gain,ai,excitation);
figure(13);
plot(synt_frame);
xlabel('time');
ylabel('amplitude');
figure(14);
periodogram(synt_frame,[],512);
```

Although the resulting waveform is obviously different from the origin alone (this is due to the fact that the LP model does not account for the phase spectrum of the original signal), its spectral envelope is identical. Its fine harmonic details, though, also widely differ: the synthetic frame is actually “over-harmonic” compared to the analysis frame (Fig. 5.13).

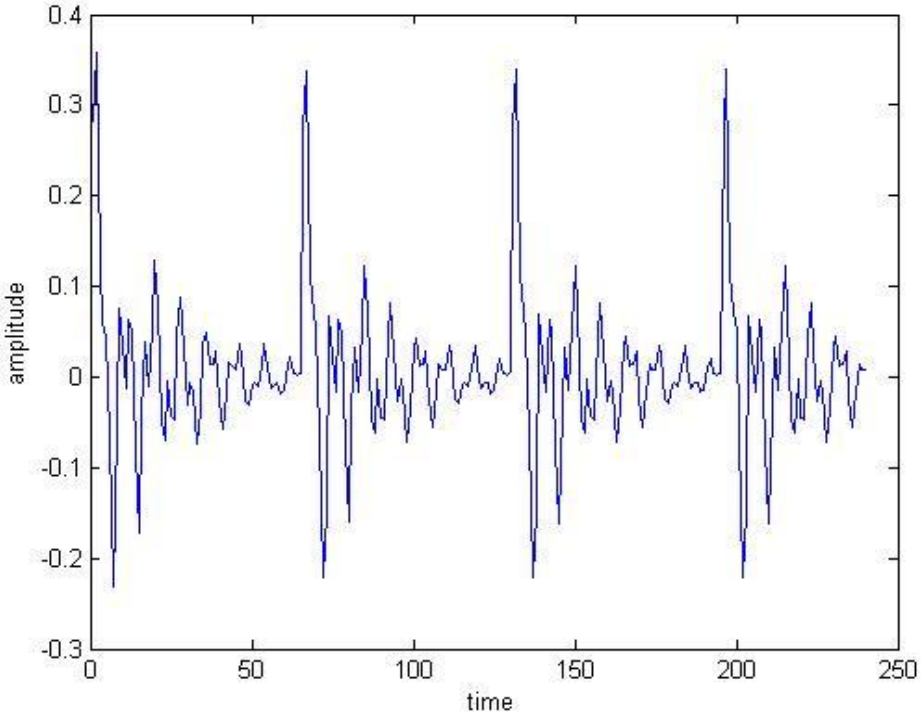


Fig. 5.13 Waveform of voiced LPC speech

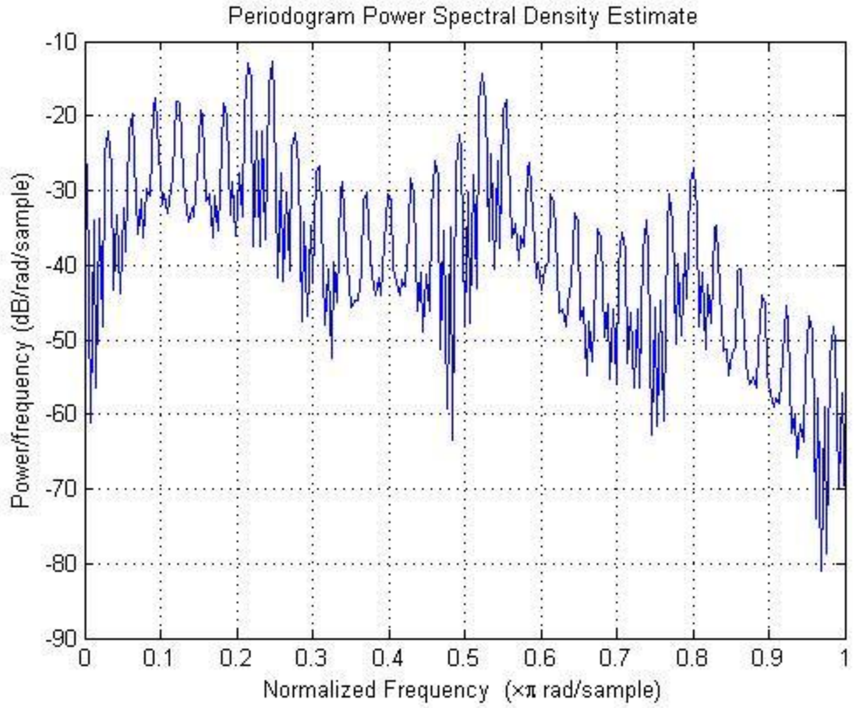


Fig. 5.14 Periodogram of voiced LPC speech

5.3 Linear prediction synthesis of 30 ms of unvoiced speech

It is easy to apply the same process to an unvoiced frame and compare the final spectra again. Let us first extract an unvoiced frame and plot it (Fig. 5.15). As expected, no clear periodicity appears.

```
speech_HF=wavread('C:\Users\Desktop\sp20_airport_sn5.wav');
input_frame=speech_HF(3500:4739);
figure(15);
plot(input_frame);
xlabel('time');
ylabel('amplitude');
```

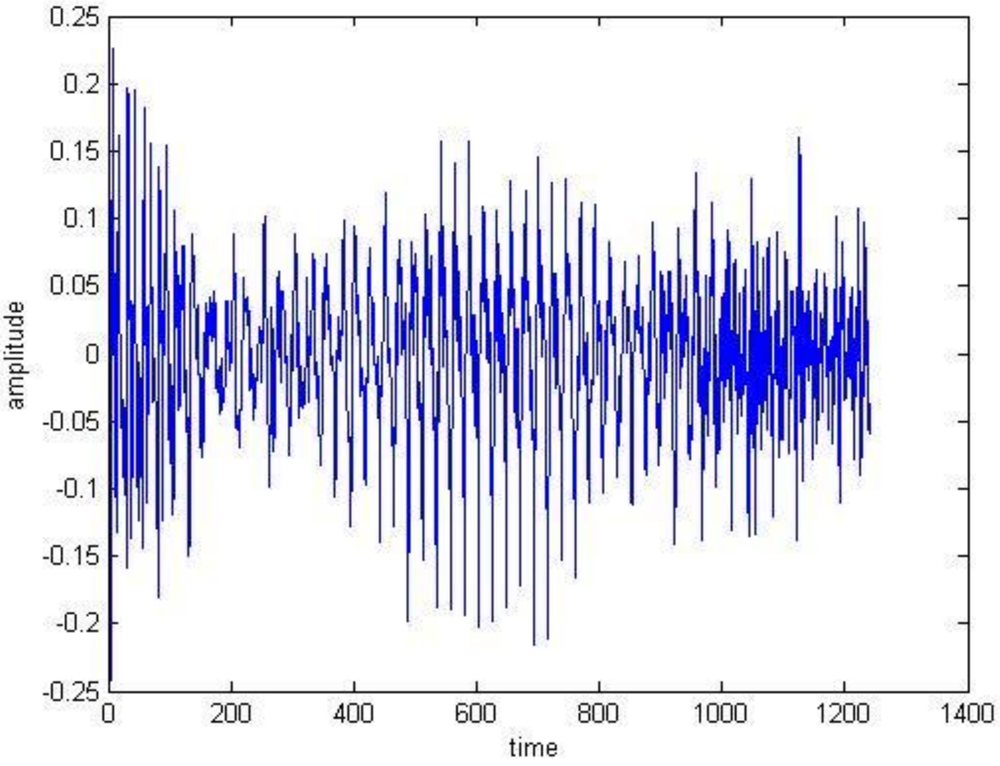


Fig. 5.15 Waveform of 30-ms-long frame of unvoiced speech

Now let us see the spectral content of this speech frame. Note that, since we are dealing with noisy signals, we use the *averaged periodogram* to estimate power spectral densities, although with less-frequency resolution than using a simple periodogram. The MATLAB pwlech function does this with eight sub frames by default and 50% overlap.

```
figure(16);
pwelch(input_frame);
```

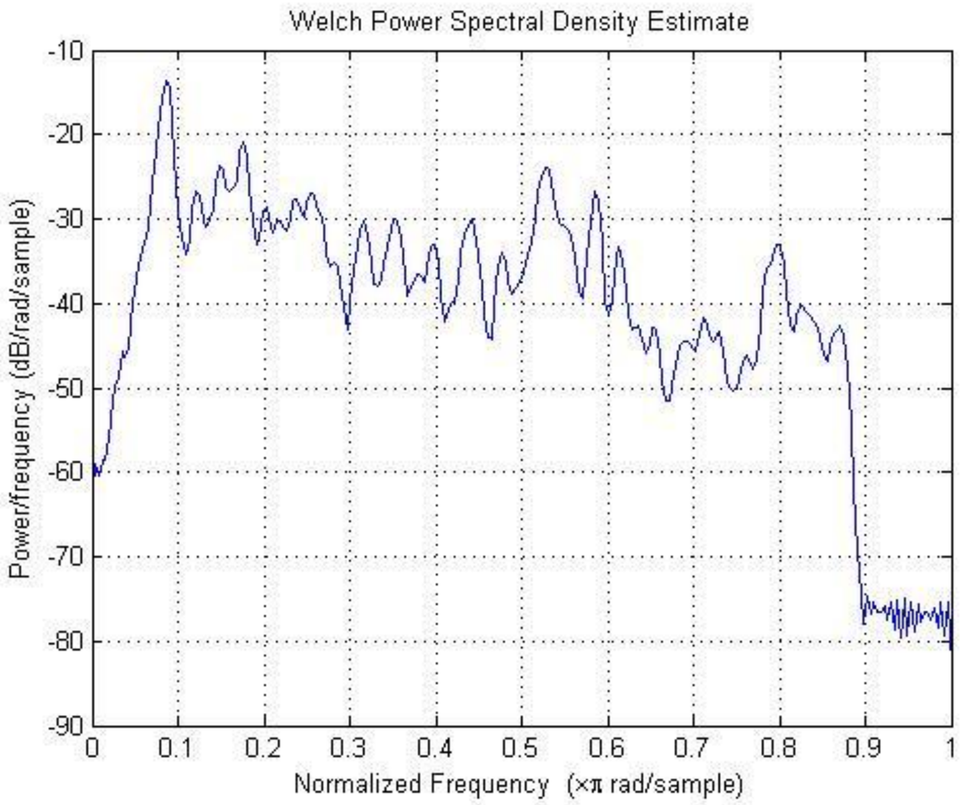


Fig. 5.16 Power spectral density of 30-ms-long frame of unvoiced speech

Let us now apply an LP model of order 10 and synthesize a new frame. Synthesis is performed by all-pole filtering a Gaussian white noise frame with standard deviation set to the prediction residual standard deviation, σ.

```
[ai, sigma_square]=lpc(input_frame,10);
sigma=sqrt(sigma_square);
excitation=randn(240,1);
synt_frame=filter(sigma,ai,excitation);
figure(17);
plot(synt_frame);
xlabel('time');
ylabel('amplitude');
figure(18);
pwelch(synt_frame);
```

Synthetic waveform (Fig. 5.17) has no sample in common with the original waveform. The spectral envelope of this frame, however, is still similar to the original one, enough at least for both the original and synthetic signals to be perceived as the same coloured noise.

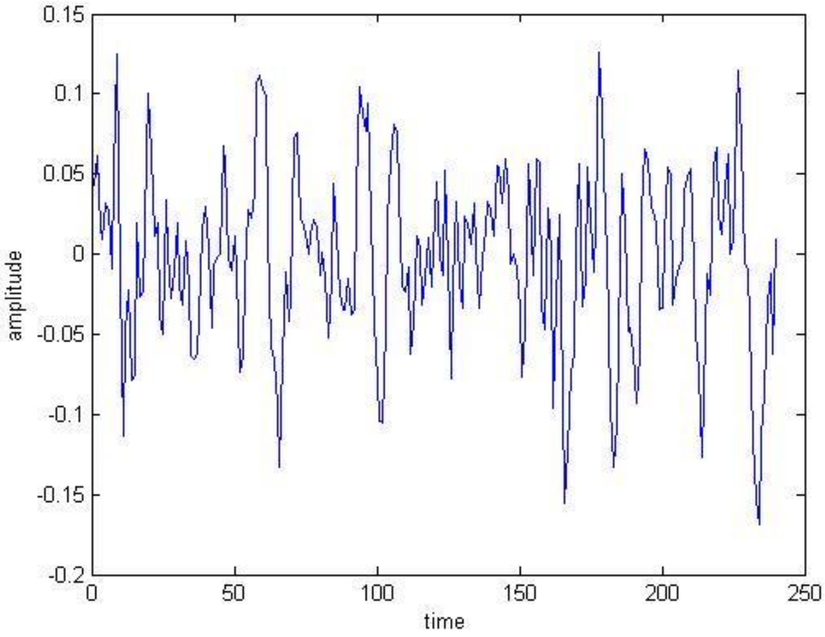


Fig. 5.17 Waveform of unvoiced LPC speech

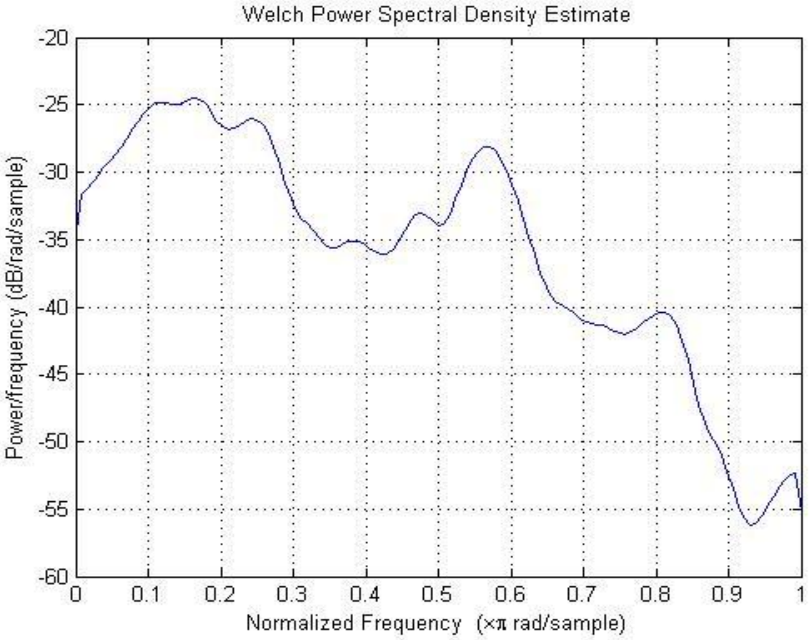


Fig. 5.18 Power spectral density of unvoiced LPC speech

CONCLUSION

Linear Predictive Coding is an analysis/synthesis technique to lossy speech compression that attempts to model the human production of sound instead of transmitting an estimate of the sound wave. Linear predictive coding achieves a bit rate of 2400 bits/second which makes it ideal for use in secure telephone systems. Secure telephone systems are more concerned that the content and meaning of speech, rather than the quality of speech, be preserved. The trade-off for LPC’s low bitrate is that it does have some difficulty with certain sounds and it produces speech that sound synthetic.

Linear predictive coding encoders break up a sound signal into different segments and then send information on each segment to the decoder. The encoder send information on whether the segment is voiced or unvoiced and the pitch period for voiced segment which is used to create an excitement signal in the decoder. The encoder also sends information about the vocal tract which is used to build a filter on the decoder side which when given the excitement signal as input can reproduce the original speech.

REFERENCES

1) Thierry Dutoit and Ferran Marqu´es, ―Applied Signal Processing A Matlab-Based Proof Of Concept,‖ Forward by Lawrence Rabiner, pp. 1–29

2) J. Ellis D (2006) Matlab Audio Processing Examples [online] Available: http://www.ee.columbia.edu/%7Edpwe/resources/matlab/ [20/2/2007]

3) Khan A, Kashif F (2003) Speech Coding with Linear Predictive Coding (LPC) [online] Available: http://www.dspexperts.com/dsp/projects/lpc [20/2/2007]

4) http://iitg.vlab.co.in/?sub=59&brch=164&sim=616&cnt=1108

5) http://my.fit.edu/~vKepuska/ece5525/lpc_paper.pdf

6) Priyabrata Sinha - Speech processing in Embedded Systems